

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
INGENIERÍA DE TELECOMUNICACIÓN



PROYECTO FINAL DE CARRERA

**MEJORA DE UN ALGORITMO DE SEGMENTACIÓN DE IRIS
MEDIANTE CONTORNOS ACTIVOS**

David Alcón Corral
Leganés, Julio 2012



PROYECTO FINAL DE CARRERA

**MEJORA DE UN ALGORITMO DE SEGMENTACIÓN DE IRIS
MEDIANTE CONTORNOS ACTIVOS**

Autor:

David Alcón Corral

Tutor/a:

Raúl Sánchez Reillo

Judith Liu Jiménez

Dpto. Tecnología Electrónica
Universidad Carlos III de Madrid

Leganés, Julio 2012

“...Pero también resultó claro que un aumento de bienestar tan extraordinario amenazaba con la destrucción era ya, en sí mismo, la destrucción de una sociedad jerárquica. En un mundo en que todos trabajaran pocas horas, tuvieran bastante que comer, vivieran en casas cómodas e higiénicas, con cuarto de baño, calefacción y refrigeración, y poseyera cada uno un auto o quizás un aeroplano, habría desaparecido la forma más obvia e hiriente de desigualdad. Si la riqueza llegaba a generalizarse, no serviría para distinguir a nadie. Sin duda, era posible imaginarse una sociedad en que la riqueza, en el sentido de posesiones y lujos personales, fuera equitativamente distribuida mientras que el poder siguiera en manos de una minoría, de una pequeña casta privilegiada. Pero, en la práctica, semejante sociedad no podría conservarse estable, porque si todos disfrutasen por igual del lujo y del ocio, la gran masa de seres humanos, a quienes la pobreza suele imbecilizar, aprenderían muchas cosas y empezarían a pensar por sí mismos; y si empezaran a reflexionar, se darían cuenta más pronto o más tarde que la minoría privilegiada no tenía derecho alguno a imponerse a los demás y acabarían barriéndoles. A la larga, una sociedad jerárquica sólo sería posible basándose en la pobreza y en la ignorancia. Regresar al pasado agrícola como querían algunos pensadores de principios de este siglo no era una solución práctica, puesto que estaría en contra de la tendencia a la mecanización, que se había hecho casi instintiva en el mundo entero, y, además, cualquier país que permaneciera atrasado industrialmente sería inútil en un sentido militar y caería antes o después bajo el dominio de un enemigo bien armado...”

George Orwell en la novela 1984. [i]

Agradecimientos

Cuando iba a empezar a escribir, no tenía mucha idea de cómo empezar, pero al mirar atrás y pensar en todos aquellos a quienes quieres dar las gracias, a quienes crees que debes una parte de esto, se siente una gran satisfacción.

Porque a pesar de lo difícil o agotador que haya podido resultar en algunos momentos entre todos habéis conseguido que mereciera la pena y que los recuerdos buenos sean los que prevalezcan.

Gracias, en primer lugar, y especialmente a mis padres y a mi hermano, quienes me han enseñado todo lo que sé, en general a toda mi familia, por la paciencia, por el apoyo, por haber confiado siempre en mí. Sin todo su apoyo nunca hubiera sido posible.

Gracias a mis amigos de Cáceres y alrededores, que aunque no han estado aquí en el día a día han sabido apoyarme cuando ha sido necesario.

Gracias a mis tutores Judith Liu Jiménez y Raúl Sánchez Reillo, por descubrirme el bonito e interesante mundo de la biometría y por haber luchado porque este proyecto saliera adelante.

Gracias a los que primero fueron compañeros de clase y de residencia, y con el paso de los años grandes amigos. Por todos esos buenos momentos y no tan buenos que pasamos juntos. Mi mejor recuerdo universitario sois vosotros.

En fin, gracias a todos los que habéis hecho que todo esto sea posible y habéis conseguido que lo lleve de forma más amena. Podría nombraros a todos, pero sabéis de sobra quiénes sois.

Gracias a todos por haber hecho que todo fuera más sencillo

Resumen

En el presente Proyecto Fin de Carrera se exploran los fundamentos de la biometría, una de las más potentes tecnologías en el ámbito de seguridad y de la medicina. Esta tecnología cada vez está más presente en la sociedad actual. En concreto se centra en el estudio de una técnica novedosa y hasta ahora poco empleada como son los contornos activos. A través de dicha tecnología el proyecto tiene como objetivo el reconocimiento biométrico del iris.

En los algoritmos biométricos por iris uno de los módulos fundamentales es el preprocesado. Este módulo se encarga de encontrar y aislar el iris dentro de una imagen. La mayor parte de los algoritmos existentes para este módulo se basan en detectar círculos dentro una imagen, asemejando el iris a una corona circular, sin embargo, en muchos casos ni el iris ni la pupila son círculos perfectos. En este proyecto se propone una mejora de estos algoritmos para que partiendo de los datos proporcionados por estos, el sistema sea capaz de detectar perfectamente la pupila y el borde exterior del iris, independientemente de la forma que estos tengan.

Para llevar a cabo este proyecto, nos basamos en varios métodos además de los *snakes*, el algoritmo de Canny, GVF, método de Otsu e imagen binaria. Todos ellos se describen teóricamente, además podemos ver un amplio estudio práctico para cada uno de los métodos utilizados a lo largo de este proyecto. Se ha implementado en C#. Se ha utilizado el programa Microsoft Visual Studio 2008.

Palabras clave: Biometría, reconocimiento de iris, snake, GVF

Índice General

I. Introducción a la Biometría

1. Introducción a la identificación biométrica

1.1. Historia	3
1.2. Funcionamiento y rendimiento	4
1.3. Funciones: Verificación vs Identificación	4
1.4. Partes de un sistema biométrico	5
1.5. Estándares asociados a tecnologías biométricas	7

II. El iris en la Biometría

2. Reconocimiento biométrico de iris

2.1. Análisis anatómico y fisiológico del ojo.	12
2.2. Historia del reconocimiento biométrico de iris	14
2.3. Fundamentos del reconocimiento biométrico de iris	15
2.4. Características del iris como rasgo biométrico	16
2.5. Etapas de un sistema de reconocimiento basado en iris	17

III. Motivaciones y objetivos. Nuestro sistema de segmentación de iris.

3. Motivaciones y objetivos. Nuestro sistema de segmentación de iris

3.1. Motivación y objetivo.	22
3.2. Nuestro sistema de segmentación de iris.	22
3.2.1. Base de datos	22
3.2.2. Diagrama de flujo	23
3.2.3. Implementación. Equipo.	24

IV. Segmentación del iris

4. Segmentación

4.1. Introducción a la segmentación de imágenes	28
4.2. Introducción a la detección de bordes	30

5. Binarización

5.1 Imagen Binaria	34
--------------------	----

5.1.1. Introducción	34
5.1.2. Resultados	34
5.1.3. Conclusión	37
5.2. Método de Otsu	37
5.2.1. Método	38
5.2.2. Algoritmo	38
5.2.3. Resultados	39
5.2.4. Conclusión	41
 6. Algoritmo de Canny	
6.1. Desarrollo del algoritmo de Canny	44
6.2. Etapas del algoritmo	44
6.3. Parámetros	49
6.4. Resultados	49
6.5. Conclusión	84
 7. Contornos Activos	
7.1. Introducción	57
7.1.1. <i>Snakes</i>	57
7.1.2. Modelo teórico tradicional	58
7.1.3. Optimización del modelo	61
7.1.4. Problemas y soluciones del modelo teórico	62
7.1.5. Comportamiento de los snakes tradicionales	65
7.2. Gradient Vector Flow (GVF)	66
7.2.1. Resultados	69
7.2.2. Conclusión	71
7.2.3. Resultados <i>snake</i>	72
7.2.4. Conclusión	82
 V. Conclusiones y líneas futuras	
8. Conclusiones y líneas futuras	
8.1. Conclusión	87
8.2. Líneas futuras	88
 VI. Presupuesto	
9. Presupuesto	
9.1. Introducción	91

Índice de figuras

1.1. Esquema de un sistema biométrico	5
2.1. Esquema del ojo humano	13
2.2. Extracto de la patente presentada por <i>L. Flor</i> y <i>A. Safir</i> en 1987.	15
3.1. Diagrama 1	23
3.2. Diagrama 2	23
3.3. Diagrama 3	23
5.1.1. Ejemplo imagen binaria	34
5.1.2.- 5.1.7. Resultados imagen binaria	34
5.2.1. Ejemplo del método de <i>Otsu</i>	39
5.2.2.- 5.2.9. Resultados del método de <i>Otsu</i>	39
5.2.10. Caso ideal del método de <i>Otsu</i>	42
5.2.11. Caso ideal del método de <i>Otsu</i>	42
6.1. Imagen usada como ejemplo en el algoritmo de Canny	45
6.2. Imagen ejemplo después de aplicar el filtro Gaussiano	45
6.3. Ejemplo de la supresión de puntos que no son máximos	47
6.4. Imagen ejemplo después de eliminar los puntos que no son máximos	47
6.5. Imagen ejemplo después de aplicar el doble umbral	48
6.6. Imagen después de aplicar el algoritmo de Canny	49
6.7. – 6.16. Resultados del método de Canny	50
7.1. Comportamiento snakes tradicionales.	65
7.2. Características snakes tradicionales.	65
7.3: Snake con fuerzas GVF	67
7.4. – 7.8. Resultados del método GVF	70
7.9. Diagrama	72
7.10. Resultado snake. Circulo	73
7.11. Resultado snake. Pentágono	73
7.12 – 7.14. Resultado preprocesado	74
7.15. – 7.22. Resultados snake para el borde interno	75
7.23. – 7.31. Resultados snake para el borde externo	79

Índice de tablas

1.1. Comparativa de distintos sistemas Biométricos	6
1.2. Principales participantes en la Industria Biométrica.	7
1.3.3.1. Características del equipo usado	24
5.1.2.1. Análisis I de los resultados	36
5.1.2.2. Análisis II de los resultados	36
5.1.2.3. Análisis III de los resultados.	36
5.1.2.4. Tiempo de ejecución.	36
5.2.1. Análisis II resultados método de <i>Otsu</i> .	42
5.1. Parámetros utilizados en el algoritmo de Canny para la figura 6.6	49
6.2. Parámetros utilizados en los resultados del algoritmo de Canny	50
6.3. Tiempos de ejecución	53
6.4. Análisis de los resultados después del algoritmo Canny	53
7.1. Parámetros utilizados en el algoritmo GVF	69
7.2. Parámetros utilizados en los resultados del algoritmo de Canny	74
7.3. Parámetros utilizados en el algoritmo GVF	74
7.4. Rango de valores para detectar el borde interno del iris mediante el snake.	75
7.5. Tiempos de ejecución	78
7.6. Análisis de los resultados después ejecutar el <i>snake</i> para el borde interno.	78
7.7. Rango de valores para detectar el borde externo del iris mediante el <i>snake</i> .	79
7.8. Tiempos de ejecución	82
7.9. Análisis de los resultados después ejecutar el <i>snake</i> para el borde externo.	82

Parte I

Introducción a la Biometría

Capítulo 1

Introducción a la Biometría

La palabra "Biometría" se deriva de las palabras griegas "bios" y "métrica", que significa "vida" y "medición" respectivamente. Esto se traduce literalmente en "Medición de la vida".

La Biometría es el estudio de métodos automáticos para el reconocimiento unívoco de personas humanas basadas en rasgos físicos o de conducta, es decir, identificar o verificar la identidad de un usuario mediante algunos de sus rasgos. La Biometría es aplicada en diferentes sectores de la ciencia. Los rasgos que se consideran incluyen las huellas dactilares, la retina, los patrones del iris, características faciales y muchos más.

La Biometría basada en rasgos físicos mide las características físicas inherentes de un individuo. Se puede utilizar tanto para la verificación, como la identificación. Hay distintos usos: Bertillonage, reconocimiento de huellas dactilares, facial, del iris, retina...

La Biometría basada en la conducta mide las características que se adquieran de forma natural durante un tiempo. Hay distintos usos: reconocimiento de voz, firma, combinación de teclas...

1.1. Historia

La Biometría no se puso en práctica en las culturas occidentales hasta finales del siglo XIX, pero era utilizada en China desde al menos el siglo XIV. Un explorador y escritor que respondía al nombre de Joao de Barros escribió que los comerciantes chinos estampaban las impresiones y las huellas de la palma de las manos de los niños en papel con tinta. Los comerciantes hacían esto como método para distinguir entre los niños y los jóvenes.

En Occidente, la identificación confiaba simplemente en la "memoria fotográfica" hasta que Alphonse Bertillon, jefe del departamento fotográfico de la Policía de París, desarrolló el sistema antropométrico (también conocido más tarde como Bertillonage) en 1883. Éste era el primer sistema preciso, ampliamente utilizado científicamente para identificar a criminales y convirtió a la Biometría en un campo de estudio. Funcionaba midiendo de forma precisa ciertas longitudes y anchuras de la cabeza y del cuerpo, así como registrando marcas individuales como tatuajes y cicatrices. El sistema de Bertillon fue adoptado extensamente en occidente hasta que aparecieron defectos en el sistema, principalmente problemas con métodos distintos de medidas y cambios de medida. Después de esto, las fuerzas policiales occidentales comenzaron a usar la huella dactilar, el mismo sistema visto en China cientos de años antes.

En estos últimos años la biometría ha crecido desde usar simplemente la huella dactilar, a emplear distintos rasgos teniendo en cuenta varias medidas físicas y de comportamiento. Las aplicaciones de la Biometría también han ido aumentando a lo largo de los últimos años.

La idea de usar patrones de iris como método de identificación fue propuesta en 1936 por el oftalmólogo Frank Burch. En 1985 los Doctores Leonard Flom y Aran Safir retomaron la idea. Su investigación y documentación les concedió una patente en 1987. En 1989 Flom y Safir recurrieron a John Daugman para crear algoritmos para el reconocimiento de iris. Estos algoritmos, patentados por Daugman en 1994¹ y que son propiedad de Iridian Technologies, son la base para gran parte de los productos de reconocimiento de iris.

En 1993 la Agencia Nuclear de Defensa empezó a trabajar con IriScan Inc. para desarrollar y probar un prototipo, 18 meses después el primer prototipo se completó y está disponible comercialmente.

1.2. Funcionamiento y rendimiento

En un sistema de Biometría típico, la persona se registra con el sistema cuando una o más de sus características físicas y de conducta son obtenidas, procesadas por un algoritmo numérico e introducido en una base de datos. Idealmente, cuando entra, casi todas sus características concuerdan; entonces cuando alguna otra persona intenta identificarse, no empareja completamente, por lo que el sistema no le permite el acceso. Las tecnologías actuales tienen tasas de acierto que varían ampliamente (desde valores bajos como el 60%, hasta altos como el 99,9%). [1]

1.3. Funciones: Verificación vs Identificación

Dependiendo de la aplicación requerida, los sistemas biométricos pueden ser usados en modo de verificación o de identificación. La verificación, también denominada autenticación, se utiliza para determinar que una persona realmente es quien dice ser. Por otro lado la identificación o reconocimiento se usa para establecer la identidad de una persona. Ninguna comparación resulta perfecta en estos sistemas, tanto de verificación como de identificación, ya que hay que garantizar que los patrones obtenidos son en todos los casos únicos, por tanto, el sistema deberá establecer un umbral que determine la tolerancia en el grado de semejanza, entre el patrón de prueba obtenido y la muestra registrada en la base de datos, siendo este umbral el que determina el límite entre el acceso permitido y el no permitido. Para ello tras la comparación de ambos patrones se lleva a cabo una medida de similaridad con el fin de obtener el grado de diferencia entre ambos en formato numérico que será comparado con el umbral predefinido para llevar a cabo la decisión oportuna.

¹John Daugman, "Iris Recognition for Personal Identification" The Computer Laboratory, University of Cambridge <http://www.cl.cam.ac.uk/~jgd1000/iris_recognition.html>

²"Segmentación de iris mediante contornos activos" Rocío Sanchiz Redondo, 2011, pag 10-15, <<http://hdl.handle.net/10016/12094>>

³"Sistema de reconocimiento de personas mediante su patrón de iris basado en la transformada Wavelet"

1.4. Partes de un sistema biométrico

El funcionamiento de un sistema biométrico puede resumirse en el esquema que se muestra en la siguiente figura:

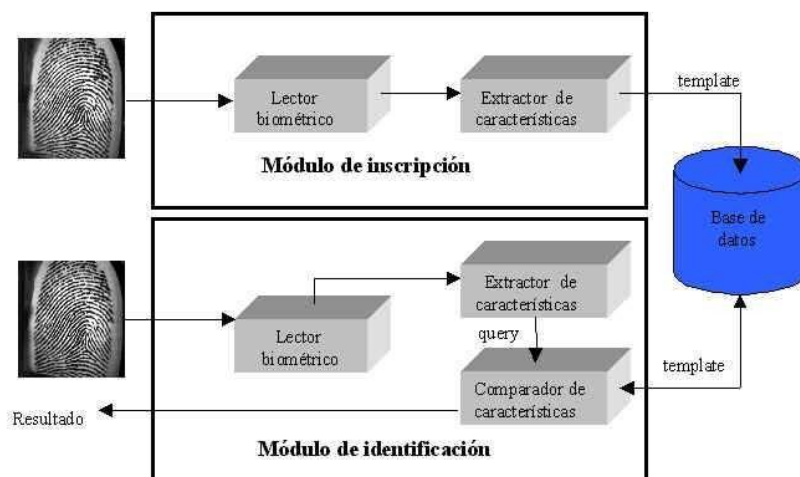


Figura 1.1: Esquema de un sistema biométrico. [2]

En él se puede observar fácilmente los pasos seguidos en el proceso de identificación:

1. En primer lugar la imagen o características biométricas son adquiridas y procesadas (extracción de las características fundamentales).
2. A continuación estas características se almacenan en una base de datos que será consultada más adelante. A estas características las denominamos patrón de usuario.
3. Finalmente en el momento en que se realiza la identificación se produce un paso similar inicial, de nuevo se captura una nueva imagen del individuo objeto de análisis y se procesa para ser comparada con el registro almacenado en la base de datos (patrón). Como resultado de esta comparación se establece si la identificación es positiva o negativa.

A partir de la observación del proceso seguido por los sistemas biométricos podemos distinguir de forma general tres bloques importantes en todos ellos:

- Un mecanismo para la obtención de datos, lógicamente se trata de una cámara o cualquier sensor para capturar la imagen (analógica o digital) o característica (voz, escritura, etc.) que intervenga en el proceso de identificación.

- El software encargado de procesar y comparar la característica biométrica a analizar.
- La interfaz con la aplicación biométrica para interactuar con los elementos relativos al acceso una vez realizada la identificación.

En el trabajo anterior a éste de Rocío Sanchiz ² o el proyecto fin de carrera de Rafael Coomonte ³ podemos encontrar un gran análisis de distintas técnicas biométricas, y la descripción de distintos mecanismos de seguridad. A continuación vemos un análisis en función a algunas de sus propiedades. Para ver un análisis más completo podemos ver sus trabajos.

	Fiabilidad	Facilidad de uso	Prevención de ataques	Aceptación	Estabilidad
Iris	Muy alta	Media	Muy alta	Media	Alta
Retina	Muy alta	Baja	Muy alta	Baja	Alta
Huellas dactilares	Muy alta	Alta	Alta	Alta	Alta
Geometría de la mano	Alta	Alta	Alta	Alta	Media
Escritura y firma	Media	Alta	Media	Muy alta	Baja
Voz	Alta	Alta	Media	Alta	Media
Cara 2D	Media	Alta	Media	Muy alta	Media
Cara 3D	Alta	Alta	Alta	Muy alta	Alta

Tabla 1.1: Comparativa de distintos sistemas biométricos. [3]

² “Segmentación de iris mediante contornos activos” Rocío Sanchiz Redondo, 2011, pag 10-15, <<http://hdl.handle.net/10016/12094>>

³ “Sistema de reconocimiento de personas mediante su patrón de iris basado en la transformada Wavelet” Rafael Coomonte Belmonte, 2006, pag 17-42, <<http://oa.upm.es/1329/>>

Tecnología	Aplicación horizontal	Principales mercados verticales
AFIS/Lifescan	Controles de Vigilancia	Servicios policiales y militares
Reconocimiento de cara	Identificación sin contacto	Farmacéuticas, Hospitales, Industria pesada y Obras
Geometría de Mano	Identificación Criminal	Hospitales y Sector Salud
Reconocimiento de iris (ojo)	Acceso a sistemas	Industria manufacturera
Reconocimiento de Voz	Acceso a instalaciones	Viajes y Turismo
Escritura y Firma	Vigilancia	

Tabla 1.2: Principales participantes en la Industria Biométrica. Las más utilizadas son el reconocimiento por huella dactilar, de cara y de iris. [1]

1.5. Estándares asociados a tecnologías biométricas

En los últimos años se ha notado una preocupación creciente por las organizaciones regulatorias respecto a elaborar estándares relativos al uso de técnicas biométricas. Esta preocupación es reflejo del creciente interés industrial por este ámbito tecnológico, y a los múltiples beneficios que su uso aporta. Sin embargo, los proveedores de soluciones biométricas realizan interfaces de software propietario para sus productos, cerrando el mercado a otros proveedores. Esta tendencia está cambiando lentamente, y se empieza a tener en cuenta la estandarización.

A nivel mundial el principal organismo que coordina las actividades de estandarización biométrica es el *Sub-Comité 37 (SC37)* del *Joint Technical Committee on Information Technology (ISO/IEC JTC1)*, de la *International Organization for Standardization (ISO)* y la *International Electrotechnical Commission (IEC)*.

En Estados Unidos desempeñan un papel similar el Comité Técnico *MI* del *INCITS (Inter National Committee for Information Technology Standards)*, el *National Institute of Standards and Technology (NIST)* y el *American National Standards Institute (ANSI)*. En Europa, este papel lo desarrolla el *CEN (Comité Europeo de Normalización)*.

Existen además otros organismos no gubernamentales impulsando iniciativas en materias biométricas tales como el *Biometrics Consortium*, *International Biometrics Groups* o *BioAPI Consortium*. Este último se estableció en Estados Unidos en 1998 compuesto por las empresas *Bioscrypt*, *Compaq*, *Iridian*, *Infineon*, *NIST*, *Saflink* y *Unisys*. El Consorcio *BioAPI* desarrolló conjuntamente con otros consorcios y asociaciones un estándar que promoviera la conexión entre los dispositivos biométricos y los diferentes tipos de programas de aplicación, además de promover el crecimiento de

los mercados biométricos.

Algunos de los estándares más importantes son:

- Estándar ANSI X.9.84: creado en 2001, por la ANSI (*American National Standards Institute*) y actualizado en 2003, define las condiciones de los sistemas biométricos para la industria de servicios financieros haciendo referencia a la transmisión y almacenamiento seguro de información biométrica y a la seguridad del hardware asociado.
- Estándar ANSI / INCITS 358: creado en 2002 por ANSI y *BioApi Consortium*, presenta una interfaz de programación de aplicación que garantiza que los productos y sistemas que cumplen este estándar son interoperables entre sí.
- Estándar NISTIR 6529: también conocido como *CBEFF (Common Biometric Exchange File Format)* es un estándar creado en 1999 por NIST y *Biometrics Consortium* que propone un formato estandarizado (estructura lógica de archivos de datos) para el intercambio de información biométrica.
- Estándar ANSI 378: creado en 2004 por la ANSI, establece criterios para representar e intercambiar la información de las huellas dactilares a través del uso de minucias. El propósito de esta norma es que un sistema biométrico dactilar pueda realizar procesos de verificación de identidad e identificación, empleando información biométrica proveniente de otros sistemas.
- Estándar ISO19794-2: creado en 2005 por la ISO/IEC con propósitos similares a la norma ANSI 378, respecto a la que guarda mucha similitud.
- Estándar PIV-071006: creado en 2006 por el NIST y el FBI en el contexto de la norma FIPS 201 del gobierno de EE.UU, establece los criterios de calidad de imagen que deben cumplir los lectores de huellas dactilares para poder ser usados en procesos de verificación de identidad en agencias federales.

Parte II

El iris en la Biometría

Capítulo 2

Reconocimiento biométrico de iris

El reconocimiento del iris es un método de autenticación biométrica que utiliza técnicas de reconocimiento de patrones (los cuales han sido almacenados anteriormente en una base de datos) en imágenes de alta resolución del iris del ojo de un individuo.

No ha de ser confundido con otro, menos frecuente, basado en el escaneo de la retina. El reconocimiento del iris utiliza la tecnología de las cámaras, con una fina iluminación infrarroja se reduce el reflejo que se haya podido producir en la convexa córnea y poder crear detalladas imágenes de las complejas estructuras del iris. Una vez convertidas en plantillas digitales, estas imágenes proporcionan una representación matemática del iris, las cuales coinciden con una identificación positiva y unívoca de un individuo.

La eficacia del reconocimiento del iris es raramente obstaculizada por gafas o lentes de contacto. La tecnología basada en el iris tiene un valor atípico pequeño (es decir, hay un número de personas reducido que no lo pueden usar) en comparación con el resto de tecnologías biométricas. Debido a su velocidad de comparación, el reconocimiento del iris es la tecnología biométrica más adecuada para la identificación de un grupo numeroso de personas. Una ventaja clave del reconocimiento del iris es su estabilidad, (el patrón o la plantilla no se modifica a lo largo del tiempo), ya que, salvo un traumatismo, esta “matrícula” individual puede durar toda la vida.

El físico John G. Daugman⁴ (Laboratorio Informático de la Universidad de Cambridge) fue pionero en este campo, desarrollando los procesos de creación de algoritmos de reconocimiento mediante el iris necesarios para la adquisición de la imagen y la puesta en el mercado de instrumentos necesarios para tal fin. Estos algoritmos se utilizaron para iniciar la comercialización de esta tecnología en conjunto con una primera versión del sistema IrisAccess diseñado y fabricado por LG Electronics, en Corea. Los algoritmos de Daugman son la base de la mayoría de los sistemas de reconocimiento del iris que se introdujeron en el mercado hasta 2006. En las pruebas, cuando los límites (umbrales) coincidían, para una mejor comparación, se cambiaba la configuración predeterminada para que permitiera una tasa de error situada entre una región de 10^{-03} a 10^{-04} a pesar de esto, los porcentajes de los códigos del iris incorrectos o rechazados son comparables con los sistemas de detección mediante la huella dactilar más precisos.

2.1. Análisis anatómico y fisiológico del ojo.

Para comprender el procedimiento de reconocimiento biométrico basado en iris es importante conocer la anatomía del ojo humano cuya fisiología juega un papel fundamental en el desarrollo de la aplicación. En primer lugar, se estudiará el ojo como órgano del cuerpo humano y las estructuras que forman parte de él, y se analizarán su misión y disposición dentro del globo ocular.

El ojo, a través de sus distintos elementos recibe los estímulos luminosos externos, los codifica y transmite a través de la vía óptica al cerebro, lugar donde se produce el fenómeno de la visión. El ojo descansa sobre una cavidad ósea, en la mitad anterior de la órbita, rodeado de músculos extra-oculares, grasa y tejido conectivo. Sólo está expuesta su parte más anterior y está protegida por el reborde orbitario óseo.

⁴<http://www.cl.cam.ac.uk/~jgd1000/>

A continuación podemos observar un esquema del ojo humano:

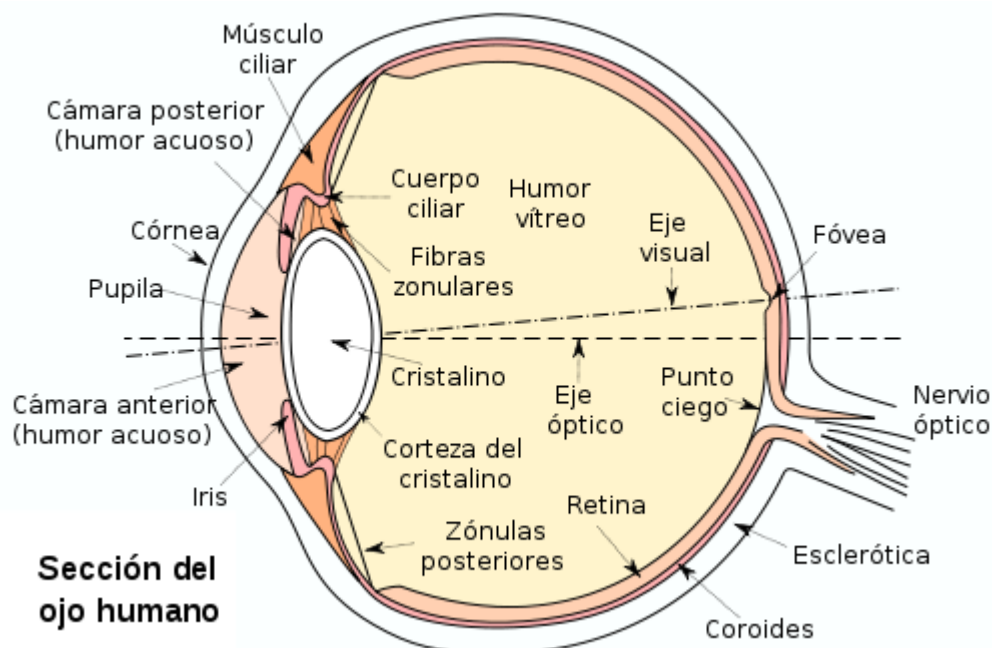


Figura 2.1: Esquema del ojo humano. [4]

El ojo está formado entre otros elementos por:

- La pupila: el círculo central, de color negro, que permite la entrada de luz al globo ocular y cuyo tamaño varía dependiendo de la cantidad de luz que llega a éste. El color negro que presenta se debe a los pigmentos retinianos.
- El iris: una membrana de color y circular cuya coloración le da la característica más visible de los ojos humanos, que es precisamente esta la diversidad de colores que presenta dependiendo del individuo. Su apertura central es la pupila. Esta membrana presenta un músculo de disposición circular que permite modificar el tamaño de la pupila.
- Un epitelio transparente, la córnea: que recubre tanto al iris como a la pupila. Esta es la primera y más poderosa lente del globo ocular, junto con el cristalino, son las lentes que permiten una visión nítida de las imágenes.
- Una zona de color blanco, la esclerótica: que forma parte de los tejidos de soporte del globo ocular. Posee función de protección y en su zona exterior está recubierta por una mucosa transparente.

El iris es el centro del análisis de este proyecto. Como ya sabemos es la membrana coloreada y circular del ojo, posee una apertura central de tamaño variable que comunica las dos cámaras (la cámara anterior y posterior): la pupila. El iris tiene como funcionalidad regular la cantidad de luz que entra en el interior del ojo, para ello varía su tamaño según la intensidad de luz que le llegue.

Las fibras musculares del iris se agrupan formando dos músculos: el esfínter del iris que produce la miosis (acción de contraer la pupila), y el dilatador de la pupila que produce la midriasis (acción de dilatar la pupila). Estas acciones se realizan de acuerdo con la intensidad que posea la fuente luminosa y tiene como objetivo regular la cantidad de luz que llega a la retina.

2.2. Historia del reconocimiento biométrico de iris

Los ojos representan una fuente fiable de patrones biométricos, tanto mediante el análisis de la retina como de los patrones característicos del iris.

Más allá de esto, el primer método biométrico basado en patrones oculares fue un sistema basado en la retina patentado por Robert Hill en 1978, que hacía uso de imágenes oftalmológicas en primer plano de los patrones internos de distribución de la formación vascular del fondo del ojo.

A pesar de que el color característico del ojo de una persona fue usado como identificador en el siglo XIX por el físico francés Alphonse Bertillon, la idea de que la compleja disposición de patrones del iris pudiera usarse como una especie de huella óptica fue propuesta por primera vez por Frank Burch, un importante cirujano ocular y oftalmólogo de St. Paul, en el año 1936, quien ya lo sugirió en uno de sus discursos, dirigido a sus colegas de profesión en el congreso anual de la American Academy of Ophthalmology.

Fue en 1987 cuando dos oftalmólogos, Leonard Flom y Aran Safir, rescataron la idea patentándola y presentándosela dos años más tarde al profesor por aquel entonces de la Harvard University John G. Daugman, con el objetivo de crear los algoritmos necesarios para desarrollar correctamente un sistema eficaz de reconocimiento de iris. Pese a su reticencia inicial a participar en el proyecto, debido a su inmediato traslado a la universidad de Cambridge, los dos oftalmólogos consiguieron atraer la atención del profesor Daugman, presentando sus ideas a través de fotografías del iris de diversos pacientes recogidas en su clínica. En estas fotografías, distintos iris presentaban complejos patrones aleatorios.

Los trabajos realizados por Daugman, que se recogen parcialmente en su trabajo titulado “High Confidence visual recognition by test of statistical independence”⁵, supusieron un avance definitivo en el campo de la identificación personal basada en patrones de iris. Los algoritmos que desarrolló posteriormente, fueron patentados en 1994⁶ (“Biométrica Personal Identification System Based on Iris Analysis”), sentando las bases de los actuales sistemas y productos de reconocimiento de iris.

Posteriormente, Aran Safira y Leonard Flor, fundaron conjuntamente con John Dignan la compañía Iris Corporación, poseedora de la patente y encargada de distribuir las licencias a las diferentes compañías desarrolladoras e integradoras de sistemas de reconocimiento que quisiesen hacer uso del patrón del iris. Una de estas empresas es Sensor Corporation que creó una cámara especial capaz de obtener imágenes de iris en los cajeros automáticos. Estas dos compañías (Iriscan Corp. y Sensor Corp.) se unieron creando Iridian Technologies.

⁵ John Daugman, “High confidence visual recognition of persons by a test of statistical independence” IEEE Transactions on Pattern Analysis and Machine Intelligence - TPAMI, vol. 15, no. 11, pp. 1148-1161, 1993

⁶ John Daugman, “Iris Recognition for Personal Identification” The Computer Laboratory, University of Cambridge <http://www.cl.cam.ac.uk/~jgd1000/iris_recognition.html>

Agencias relacionadas con los cuerpos de seguridad y justicia de los Estados Unidos comenzaron a utilizar este tipo de dispositivos en 1994, siendo la prisión del estado de Lancaster (Pennsylvania) la pionera en el uso de esta tecnología para la identificación.

United States Patent [19]		[11] Patent Number:	4,641,349
Flom et al.		[45] Date of Patent:	Feb. 3, 1987
[54] IRIS RECOGNITION SYSTEM			
[76] Inventors: Leonard Flom, 1903 Post Rd., Fairfield, Conn. 06403; Alan Safir, 3 Ellsworth Ave., Cambridge, Mass. 02130			
[21] Appl. No.: 703,312			
[22] Filed: Feb. 20, 1985			
[51] Int. Cl.: G06K 9/00			
[52] U.S. Cl.: 382/2; 351/205; 351/206; 354/62; 382/6			
[58] Field of Search: 382/2, 6; 351/206, 221, 351/208, 205; 354/62; 382/227, 231			
[56] References Cited			
U.S. PATENT DOCUMENTS			
1,192,512	5/1916	Fitz	351/211
1,887,115	11/1932	Brant	351/221
2,434,880	1/1948	Senders	354/214
3,136,139	6/1964	Safir	351/211
3,346,786	1/1968	Dolan	382/231
3,473,868	10/1969	Young	351/210
3,482,904	12/1969	Volz	351/205
3,533,683	10/1970	Stark	351/206
3,533,684	10/1970	Stark et al.	351/206
3,590,107	8/1971	Idokawa	351/209
3,600,090	8/1971	Mohanan	354/153
3,776,135	12/1973	Diamiti	351/221
3,915,564	10/1975	Urban	351/206
3,936,844	2/1976	Matterson	354/62
3,966,310	4/1976	Lancon	351/221
4,007,980	2/1977	Bracher	351/219
4,008,606	2/1977	Talbotson	354/64
4,023,037	5/1977	Weiss	250/313
4,106,237	8/1978	Hill	382/2
4,136,714	2/1979	Trachiloud	382/231
4,172,632	10/1979	Holmes, Jr.	250/312
4,186,115	2/1980	Humphrey	351/39
4,233,654	11/1980	Trachiloud	382/231
4,252,743	3/1981	Matsunaga	354/62
4,257,667	3/1981	Kobayakawa	354/62
4,266,861	5/1981	Sawa	351/206
4,308,082	1/1982	Morrison	351/39
4,375,330	3/1983	Seimand	351/212
4,392,366	7/1983	Hill	382/2

4,523,820 6/1985 Kaskas 351/206
4,533,222 8/1985 Idokawa 351/206

OTHER PUBLICATIONS

The Uveal Tract and Its Endogenous Inflammation, Hans Remky, editor, vol. 5, No. 3, Sep. 1965, pp. 630-633.
Outer Sign in Slit-Lamp Microscopy, James H. Doggart, p. 27, (London: 1949).
"Diseases of the Uveal Tract," Duke-Elder and Perkins, in *System of Ophthalmology*, Duke-Elder, Sir Stewart, p. 3 (St. Louis: 1966).
J. Hecht, "Light Modulators Help Crunch Image Data", *High Technology*, (Jan. 1983), pp. 69-72.
C. Simon & I. Goldstein, "A New Scientific Method of Identification", vol. 33, No. 18, *State Journal* (Sep. 1935), pp. 901-906.

Primary Examiner—Leo H. Boudress
Assistant Examiner—Joseph Mancuso
Attorney, Agent, or Firm—Ostrosnik, Faber, Gerb & Soffen

[57] ABSTRACT

Methods and apparatus are disclosed for identifying an eye, especially a human eye, on the basis of the visible features of the iris and pupil. The eye is first illuminated until the pupil reaches a predetermined size, at which an image of the iris and pupil is obtained. This image is then compared with stored image information for identification. The stored image information is previously obtained from an eye, the pupil of which was similarly brought to the same predetermined size. The illumination of the iris may include oblique illumination from several positions around the circumference of the iris. The illumination from each position may be relatively monochromatic, so that the resulting shadow will lack the color of the light source at that position, providing better contrast for elevation-dependent features. A system for performing iris recognition may include a processor which controls an illumination control circuit and a camera to obtain images at several predetermined sizes of the pupil.

32 Claims, 12 Drawing Figures

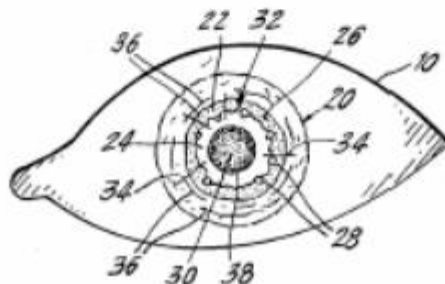


Figura 2.2: extracto de la patente presentada por L. Flor y A. Safir en 1987.

2.3. Fundamentos del reconocimiento biométrico de iris

Como hemos comentado al comienzo de este capítulo, el iris es una estructura muscular que adapta la apertura de la pupila dependiendo de la cantidad de luz que llega, y cuyos singulares detalles le dan un carácter peculiar.

El reconocimiento por iris se basa precisamente en los detalles que presenta su textura, es decir, características inherentes a su morfología, grietas, surcos o estrías entre otras características que conforman una textura altamente rica en detalles. Esta textura formada en la etapa embrionaria es estocástica y posiblemente de generación caótica, es

decir, que diferentes individuos presenten iris del mismo genotipo, conjuntos como gemelos idénticos o siameses, estos individuos deben tener detalles no correlacionados.

El iris, por tanto, presenta ciertas características especiales que le confieren un gran potencial para su aplicación en los sistemas biométricos. La cantidad de información que presenta este indicador biométrico es tan alta que permite la identificación de individuos a través de procedimientos no invasivos y que adicionalmente se desarrolla a distancias prudenciales y sin restricciones ambientales con resultados muy seguros. Esto permite la implementación de sistemas de verificación e identificación aplicables a condiciones reales. En los sistemas reales se debe dar gran importancia a los algoritmos de preprocesado y procesamiento como tal, debido a que la información efectiva con que se cuenta para lograr el proceso de reconocimiento de identidad es mucho menor que la que se obtiene en sistemas similares bajo condiciones controladas de prueba y evaluación. Entre los sistemas de reconocimiento de identidad existentes en la actualidad, son pocos los procedimientos no invasivos. En el mundo actual, donde las aplicaciones de seguridad son cada día más necesarias resulta de vital importancia el desarrollo de técnicas cuya implantación no represente incomodidad para los usuarios, ni supongan peligro para la integridad de las personas.

2.4. Características del iris como rasgo biométrico

El iris, al igual que la vasculatura retinal, posee una estructura única por individuo formando un sistema muy complejo, de modo que la posibilidad de encontrar dos iris idénticos es de 1 entre 16 millones [5]. Además se mantiene inalterable durante toda la vida de la persona, y lógicamente es éste el factor principal que propicia el que el iris sea un rasgo de elevadas prestaciones en los sistemas biométricos. Adicionalmente encontramos el factor de que esta estructura contiene gran cantidad de información, muy propicia para el análisis biométrico ya que contiene alrededor de 266 características distintivas, entre las que se encuentra el retículo trabecular, que conforma un tejido que da la sensación de dividir el iris radialmente, así como estrías, anillos, surcos, pecas y la corona característica.

A continuación enumeraremos con mayor detalle algunos de los factores que hacen que el iris sea idóneo como objeto de identificación biométrica:

- La invariabilidad: el patrón del iris se mantiene sin cambios. No se degrada con el tiempo o con el ambiente gracias a la protección que le confiere la córnea. Esto supone que el patrón que se almacenó inicialmente puede ser utilizado durante toda la vida, ya que este es estable desde aproximadamente los dieciocho meses de edad.
- La cornea: gracias a su transparencia permite hacer visible el iris desde el exterior. Es el único órgano interno que posee esta característica.

- Unicidad: Los patrones del iris son más complejos y aleatorios que otros patrones biométricos, lo cual ofrece un método de alta precisión para la autenticación individual de cada uno de los usuarios, con una tasa de error por falsa aceptación inferior a uno sobre 1.2 millones.
- Una característica importante para evitar posibles falsificaciones consiste en que, incluso con iluminación uniforme, el iris presenta pequeñas variaciones en su apertura. Esta característica es explotada en sistemas de identificación ya que permite captar si el sujeto detectado es un sujeto vivo, evitando posibles fraudes, por ejemplo al presentar una fotografía al sistema.
- Se trata de un método de identificación biométrica no invasivo. El usuario debe situarse frente al sistema de captura pero no es necesario mantener contacto con él.
- Sencillez: El estudio de patrones se hace con técnicas habituales de tratamiento de imágenes, lo que le confiere ventaja frente a la mayoría de técnicas de Biometría modernas que requieren complejas reconstrucciones en 3D.
- El sistema es invariable al uso de lentes de contacto o gafas, ya que estos elementos no modifican la estructura del ojo. También se trata de un sistema invariable a las lentes de contacto de colores.
- Presentan elevadas tasas de reconocimiento.

La cantidad de información que puede ser medida en el iris es significativamente mayor que la que puede obtenerse de las huellas dactilares. Y aunque su precisión es menor que la que presenta el ADN, la identificación del iris se considera un método rápido y preciso.

En un sistema que ha de funcionar bajo condiciones físicas reales deben ser tomadas en cuenta ciertas consideraciones: dos imágenes del mismo iris pueden estar ubicadas de manera diferente en la imagen, con pequeñas rotaciones entre ambas, elementos como pestañas y párpados pueden interferir en la información efectiva del iris, siendo éste un parámetro muy importante a considerar, sobre todo en nichos de población como el asiático, las variaciones de la imagen introducidas por las fuentes de luz, que pueden ser disminuidas con los sistemas de adquisición actuales. Todas ellas pueden representar dificultades sustanciales en el proceso de verificación de identidad.

2.5. Etapas de un sistema de reconocimiento basado en iris.

Las etapas que forman parte del proceso de identificación biométrica para el caso concreto del iris son las siguientes:

1. La primera de éstas sería, lógicamente, la fase de adquisición de la imagen del iris.

2. Posteriormente se lleva a cabo una etapa de pre-procesado de la imagen, en la que se localiza y aísla la estructura del iris, detectando sus límites exterior e interior (frontera con la esclerótica y la pupila respectivamente). Es en esta etapa del proceso en la que se centra este proyecto, determinando los límites interior y exterior del iris mediante la técnica de los contornos activos (*snakes*) que se detallará más adelante.
3. El paso siguiente será realizar la extracción de características de la imagen del iris pre-procesada, con el objetivo de determinar un vector de características, que posteriormente será comparado con el patrón almacenado en la fase de reclutamiento.
4. El sistema de reconocimiento finalizará con la etapa de comparación, en la que se pueden aplicar diferentes algoritmos, como por ejemplo los basados en la mínima distancia entre el patrón de almacenado y los modelos obtenidos en cada una de las capturas realizadas cuando un individuo utiliza el sistema.

Parte III

Motivaciones y objetivos

Nuestro sistema de segmentación de iris.

Capítulo 3

Motivaciones y objetivos

Nuestro sistema de segmentación de iris.

3.1. Motivaciones y objetivo

El estudio a realizar, dentro del ámbito del reconocimiento de iris, se centrará en la segmentación de la imagen para la obtención de la sub-imagen que sea únicamente iris.

La extracción de características y especialmente en el ámbito del reconocimiento de iris, la segmentación de iris revierte un especial interés debido a la importancia de optimizar el sistema para una correcta localización y aproximación de la característica objeto de estudio en la identificación o verificación de personas. Disminuye el tiempo de computación y, sin llegar a ser ligera, mejora eficientemente el sistema para su uso en tiempo real, al eliminar partes que realmente no conforman el iris. Es decir, evita la adquisición de componentes no deseadas en el modelo y facilita la labor de las sucesivas fases.

Con dicho objetivo, en este proyecto se ha desarrollado un algoritmo de segmentación de iris que puede emplearse tanto en procesos de reconocimiento biométrico como en otro tipo de aplicaciones, como por ejemplo pueden ser pruebas medicas (existen en la actualidad, pruebas de análisis de imagen oftalmológica en cuyo proceso interviene la segmentación de iris para detección de glaucoma), mecanismos de detección de sujeto vivo. etc.

La motivación para el desarrollo de este algoritmo se debe a que los sistemas más empleados hasta la fecha se han basado en la detección de círculos perfectos. Como bien sabemos la pupila y el iris no han de ser perfectamente circulares por lo que el hecho de desarrollar un método capaz de adaptarse a la forma real del objeto que pretendemos detectar (en este caso el iris) y hacerlo con cierta exactitud supone un gran adelanto a la vez que abre nuevos caminos de investigación.

En sucesivos capítulos se analizará en detalle el algoritmo desarrollado que nos permite detectar el iris con mayor flexibilidad y adaptabilidad, así como los fundamentos matemáticos en que se basa.

3.2. Nuestro sistema de segmentación de iris.

3.2.1. Base de datos

Para realizar nuestro sistema de segmentación de iris, se ha utilizado un lenguaje de alto nivel orientado a objetos, como es C#. Se ha empleado una base de datos, perteneciente al proyecto ICE 2005 ⁷. ICE 2005 fue un proyecto para desarrollar tecnología para el reconocimiento de iris. Este trabajo se llevó a cabo entre agosto de 2005, y marzo de 2006. El objetivo principal de este trabajo era promover e impulsar esta tecnología, para apoyar los esfuerzos del gobierno de los EEUU en desarrollarlo. Tiene un total de 2954 imágenes que se han utilizado para realizar diferentes pruebas a nuestro sistema, hasta llegar a un sistema eficiente. De ellas 1528 son del ojo izquierdo perteneciente a 120 personas, y 1425 del ojo derecho perteneciente a 124 personas. De 112 personas se toman imágenes tanto del ojo izquierdo, como del derecho. Hay un total de 132 personas diferentes.⁸

⁷[ICE - Iris Challenge Evaluation](http://ice.ece.nyu.edu/)

⁸http://biometrics.nist.gov/cs_links/iris/ice/ICE_2005_Results_30March2006.pdf

3.2.2. Diagrama de flujos

A continuaci3n vamos a mostrar diferentes diagramas en el que se muestran las diferentes etapas de nuestro sistema. El planteamiento inicial de nuestro proyecto se puede ver en el siguiente esquema:

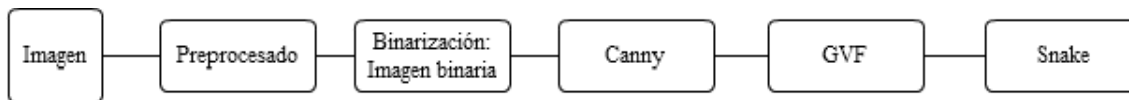


Figura 3.1: Diagrama 1

La primera etapa es la obtenci3n de la imagen, en nuestro caso disponemos de una base de datos, por lo que la obtenemos a partir de la ruta de la imagen. Posteriormente se lleva a cabo una etapa de pre-procesado de la imagen, en la que se localiza y aísla la estructura del iris, detectando sus límites exterior e interior (frontera con la escler3tica y la pupila respectivamente).

Las siguientes etapas son diferentes algoritmos de segmentaci3n de imágenes con el objetivo de lograr determinar los límites interior y exterior del iris mediante distintas técnicas que se describe en los siguientes capítulo. A partir de aquí hemos utilizado para la binarizaci3n distintos métodos hasta conseguir resultados satisfactorios. Inicialmente hemos realizado un proceso de binarizaci3n manual, que íbamos ajustando para cada imagen con el fin de encontrar unos valores que pudiéramos generalizar para toda la base de datos. No hemos encontrado estos valores, por lo que hemos intentado aplicar un ajuste automático de la binarizaci3n mediante el método de Otsu.

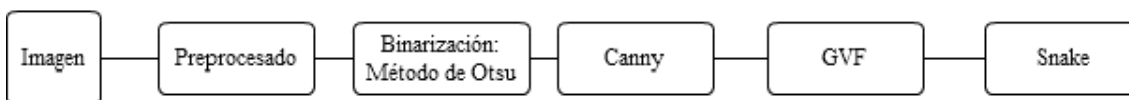


Figura 3.2: Diagrama 2

Se han realizado pruebas con este algoritmo, se ha conseguido ajustar el método correctamente, pero con nuestra base de datos no se han conseguido resultados satisfactorios.

Finalmente, tras muchas pruebas con Canny, se ha decidido aplicar directamente este método ya que se consiguen resultados realmente buenos.

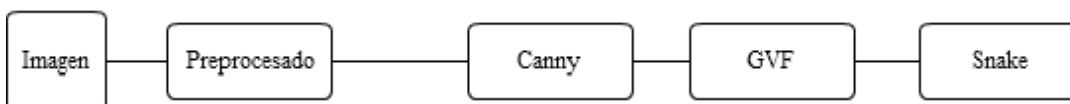


Figura 3.3: Diagrama 3

Tras Canny, se ha aplicado los métodos de Gradiente Vector Flow, y mediante el *snake* se ha detectado los límites exterior e interior. Se hace una descripci3n detallada de cada algoritmo utilizado en los siguientes capítulos.

Cada una de las etapas se ha implementado independientemente para poder tener un entorno de pruebas para cada método. De esta forma se pueden ajustar los parámetros,

sin que el resto de métodos afecten directamente. La descripción teórica de cada uno de los métodos se detalla en los siguientes capítulos.

3.2.3. Implementación y equipo

Cómo hemos comentado anteriormente se ha utilizado un lenguaje de alto nivel orientado a objetos, como es C#. Se ha decidido utilizar este lenguaje, ya que conocíamos las limitaciones en cuanto a tiempos de ejecución de otros lenguajes como Matlab.

Las etapas de binarización de Canny y de GVF, se han implementado sin el uso de librerías de terceros. Se ha hecho únicamente uso del MSDN de Microsoft para C#. Se consideró la posibilidad de usar Emgu CV, pero finalmente, con las librerías del MSDN fue suficiente.

Emgu CV es una plataforma cruzada que se creó a partir de OpenCV para el procesamiento de imágenes. Permite que las funciones de OpenCV puedan ser llamadas desde .NET. Los lenguajes compatibles son algunos como: C#, VB, VC ++.

En la última parte de nuestro algoritmo, la etapa del *snake*, sí se ha hecho uso de la librería Emgu CV. Todas las pruebas se han realizado con la versión opensource, pero dispone de versión con licencia comercial.⁹

Para las pruebas se ha utilizado el siguiente equipo:

Acer Aspire 5943G
Windows 7 Home Premium 64 bits
Intel Core i7-720QM (1.6 GHz L3)
4 GB DDR RAM

Tabla 3.1: Características del equipo usado

⁹EmguCV - http://www.emgu.com/wiki/index.php/Main_Page

Parte IV

Segmentación del iris

Capítulo 4

Segmentación

4.1. Introducción a la segmentación de imágenes

La aparición de la tecnología multimedia y el abaratamiento de las herramientas digitales, como cámaras digitales y escáneres, así como la reducción de coste del almacenamiento masivo ha dado lugar a grandes colecciones de medios audiovisuales, tanto con derechos libres, como con propiedad en Internet. Actividades como la fabricación, la medicina, el ocio, la educación, la video vigilancia, proyector I+D, etc. hacen uso de inmensas cantidades de datos audiovisuales y según la cantidad de información en formato digital aumenta, la necesidad del desarrollo de herramientas para una efectiva transformación, almacenamiento, gestión, búsqueda, filtrado y recuperación de la misma se hace evidente.

Una gran cantidad de información visual está a disposición de profesionales en forma de imágenes fijas, ya sea mediante un gráfico o mediante una fotografía, que a su vez puede ser una imagen real o no real (esbozos a mano o sintetizadas por ordenador). Estas imágenes pueden guardarse en bases de datos tanto locales como distribuidas, y añadirse a otros tipos de documentos como ficheros de texto.

Más concretamente el proceso digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales cuya finalidad es mejorar la calidad o facilitar la búsqueda de información. Esto se realiza gracias a procesos de filtrado con lo que a partir de una imagen origen, se obtiene otra final más adecuada para una aplicación específica donde ciertas características han sido mejoradas para realizar el procesado sobre dicha imagen.

Los objetivos del filtrado son: suavizar la imagen (reducir la cantidad de variaciones de intensidad entre píxeles vecinos), eliminar ruido (eliminar aquellos píxeles en los que el nivel de intensidad es muy diferente al de sus vecinos), realzar bordes (destacar los bordes que se localizan en una imagen), o detectar bordes (detectar los píxeles donde se produce un cambio brusco en la función intensidad). Por lo que podríamos decir que los filtros realizan operaciones a nivel de píxel en una imagen para optimizarla, y conseguir información relevante, o conseguir un efecto especial en ella.

Hay dos formas de filtrados: en frecuencia o en espacio.

- En el filtro de frecuencia se trabaja sobre el dominio de la frecuencia en la Transformada de Fourier de la imagen, habiendo aplicado previamente el Teorema de la convolución correspondiente. Puede ser, de paso bajo, paso alto, o paso banda.
- En el filtro del espacio se trabaja directamente sobre los píxeles de la imagen. En este proceso se relaciona todos los píxeles de una imagen próximos al píxel objetivo con la finalidad de conseguir información útil. Puede filtrarse con filtros lineales o no lineales. Hay también varios tipos, paso bajo, paso alto,

realce de bordes por desplazamiento y diferencia, realce de bordes mediante Laplace, realce de bordes con gradiente direccional, y detección de bordes y filtros de contorno (Prewitt y Sobel). [6]

La segmentación es el primer paso a realizar en cualquier aplicación de análisis de imágenes. Es de gran importancia que en esta primera toma de contacto con la imagen se obtengan resultados fiables ya que serán los cimientos sobre los que después se construirá todo el edificio, es decir, la aplicación.

La segmentación en el campo de la visión por ordenador es el proceso de dividir una imagen digital en varios grupos de píxeles u objetos, más concretamente es el proceso de asignación de una etiqueta a cada píxel de la imagen de manera que los píxeles que compartan la misma etiqueta tendrán algunas características similares, como el color, la intensidad, o la textura. Regiones adyacentes son diferentes con respecto a la misma característica. Su objetivo es simplificar o cambiar la imagen en otra más significativa y fácil de analizar. El resultado de la segmentación es un conjunto de segmentos que cubren en conjunto a toda la imagen, o conjunto de contornos extraídos de la imagen (detección de bordes). [6]

Los principios de los algoritmos de segmentación se basan en:

- Discontinuidad de niveles grises: consiste en segmentar la imagen a partir de los cambios bruscos en los niveles de gris entre los píxeles. Algunas técnicas que lo utilizan como principio son la detección de bordes, de líneas, puntos aislados.
- Similitud de niveles de grises: es lo contrario a lo anterior, las divisiones de la imagen se hacen agrupando los píxeles que tienen características similares. Algunas técnicas que lo utilizan es la umbralización, el crecimiento de regiones....

A pesar de los diferentes intentos de clasificación, elegir el mejor de los métodos es una tarea difícil, sin embargo, el pensamiento generalizado acerca de la división de una imagen parece ser la que más se acerque a la realidad. Algunos autores los dividen en los siguientes grupos: basados en regiones, que dependen de la homogeneidad de las características localizadas espacialmente (color, textura,...), y basados en contornos, que utilizan información del gradiente para localizar bordes de las distintas áreas.

Algunas de las áreas de aplicación que figuran con más frecuencia, donde el uso y recuperación de imágenes hoy en día desempeña un papel fundamental, son: la ley y prevención del delito, la medicina, dentro de las cuales estarían las pruebas médicas (localización de tumores y otras patologías, medida de volumen de tejido, cirugía asistida por ordenador, diagnóstico, planificación del tratamiento, estudio de la estructura anatómica), la moda y el diseño gráfico, editoriales, comercio electrónico, arquitectura e ingeniería de diseño, investigación histórica, como interfaz de usuario, localización de objetos en imágenes por satélites, reconocimiento de objetos, sistemas de control de tráfico, visión por computador, la literatura, fumigación de cultivos, etc.

Un estudio detallado de imágenes y los posibles usos de éstas ha sido realizado por Eakins y Graham¹⁰.

Como conclusión, debido a todo lo mencionado anteriormente existe una clara necesidad de clasificadores o de motores de búsqueda en cualquier imagen, ya sea para particulares o uso profesional.

4.2. Introducción a la detección de bordes

La detección de bordes es una herramienta fundamental en el procesamiento de imágenes y visión por computador. Estas herramientas están destinadas a la identificación de puntos en una imagen digital en el que los cambios bruscos en una imagen son muy relevantes, es decir, tiene discontinuidades. Está orientado sobre todo en las áreas de detección de características y extracción de características. El mismo problema de encontrar discontinuidades en señales 1D se conoce como la detección de paso.

El principal objetivo de la detección de bordes es el de reducir significativamente la cantidad de datos en una imagen, a la vez que mantiene las principales propiedades de la imagen, para así poder utilizarla en el procesamiento de imágenes.

Motivaciones

El propósito de detectar cambios bruscos en el brillo de la imagen es capturar eventos y cambios importantes en las propiedades de la imagen. Se puede demostrar que, bajo supuestos bastante generales para un modelo de formación de la imagen, las discontinuidades en el brillo de la imagen probablemente correspondan a¹¹ [7]:

- discontinuidades en profundidad,
- discontinuidades en la superficie, la orientación
- cambios en las propiedades del material y
- variaciones en iluminación de la escena.

En el caso ideal, el resultado de aplicar un detector de bordes en una imagen puede conducir a un conjunto de curvas conectadas que indican los límites de los objetos, los límites de las marcas superficiales así como las curvas que corresponden a discontinuidades en la orientación de la superficie. Así, aplicando un algoritmo de detección de borde a una imagen puede reducir significativamente la cantidad de datos a procesar y por lo tanto, se puede filtrar la información que pueda ser considerado como menos relevante, preservando al mismo tiempo las propiedades estructurales importantes de una imagen. Si la detección de borde tiene éxito, la tarea posterior es la interpretación de la información contenida en la imagen original, que debería

¹⁰J. P. Eakins and M. E. Graham, "Content-Based Image Retrieval: A Report To The Jisc Technology Application Programme," Institute For Image Data Research, University Of Northumbria At Newcastle, Uk, Technical Report, 1999.

¹¹H.G. Barrow and J.M. Tenenbaum (1981) "Interpreting line drawings as three-dimensional surfaces", Artificial Intelligence, vol 17, issues 1-3, pages 75-116.

simplificarse considerablemente. Sin embargo, no siempre es posible obtener los bordes ideales a partir de imágenes relativamente complejas de la vida real. Los bordes extraídos en imágenes no triviales se ve a menudo dificultada por la fragmentación, lo que significa que las aristas no están conectadas, la falta de segmentos de borde, así como los bordes falsos que no corresponden a fenómenos interesantes en la imagen complica la tarea posterior de la interpretación de los datos de la imagen. [2.5.4]

La detección de bordes es una de las etapas fundamentales en el procesamiento de imágenes, análisis de imágenes, reconocimiento de patrones de imagen y las técnicas de visión por ordenador.

Propiedades del borde

Los bordes extraídos de una imagen bidimensional, que a su vez puede ser tomada en un plano tridimensional, puede clasificarse como un punto de vista dependiente o un punto de vista independiente. Un borde punto de vista independiente refleja típicamente propiedades inherentes de los objetos tridimensionales, tales como marcas de la superficie y forma de la superficie. Un punto de vista extremo dependiente puede cambiar a medida que cambia el punto de vista y por lo general refleja la geometría de la escena, como la obstrucción entre objetos.

Un borde típico podría ser, por ejemplo, la frontera entre un bloque de color rojo y un bloque de color amarillo será una línea entre ellos. Para una línea hay generalmente un borde en cada lado de la línea.

Un modelo simple de borde

Se ha tenido en cuenta en la detección de bordes los modelos ideales, pero en ocasiones los bordes obtenidos a partir de imágenes reales no son modelos completamente ideales para la detección de bordes. Se ven afectados normalmente por uno o varios de los siguientes efectos:

- Desenfoque focal causado por una profundidad de campo y la función de dispersión puntual limitadas.
- Penumbra de imágenes borrosas causadas por las sombras.
- Sombreado de un objeto liso.

Un número de investigadores han utilizado un borde suavizado por un filtro *Gaussiano* como una simple extensión del modelo de borde ideal para modelar los efectos que desenfocan los bordes en aplicaciones prácticas.¹² [2.5.4] Así, una imagen unidimensional f que tiene exactamente un borde situado en el punto $x=0$ puede ser modelado como:

$$f(x) = \frac{I_r - I_l}{2} \left(\operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) + 1 \right) + I_l.$$

¹²W. Zhang and F. Bergholm (1997) "Multi-scale blur estimation and edge type classification for scene analysis", International Journal of Computer Vision, vol 24, issue 3, Pages: 219 - 250.

En la parte izquierda del borde, la intensidad es $I_l = \lim_{x \rightarrow -\infty} f(x)$, ya la derecha del borde es $I_r = \lim_{x \rightarrow +\infty} f(x)$. El parámetro de escala σ se llama la escala de desenfoque del borde.

Enfoques

Hay muchos modelos para la detección de bordes, pero la mayoría de ellos se pueden agrupar en dos categorías, basadas en búsquedas y cruces por cero. Los métodos de detección de bordes basados en la búsqueda primero calcula la resistencia de los bordes, normalmente una derivada de primer orden como la magnitud del gradiente, y luego la búsqueda de máximos locales de la magnitud del gradiente utilizando una estimación calculada de la orientación local del borde, normalmente la dirección del gradiente. La búsqueda y cruces por cero es un método que calcula la derivada de segundo orden a partir de la imagen con el fin de encontrar los bordes, normalmente los cruces por cero de la función *laplaciana* o los cruces por cero de una expresión diferencial no lineal. Como una etapa de pre-procesamiento para la detección de bordes, una etapa de suavizado, típicamente suavizado *Gaussiano*, casi siempre se aplica.

Los métodos de detección de bordes que han sido publicados se diferencian principalmente en los tipos de filtros que se aplican para suavizar y la forma en que se mide la resistencia de los bordes. Muchos de los métodos de detección de bordes se basan en el cálculo de los gradientes de la imagen, sin embargo, se diferencian en los tipos de filtros utilizados para las estimaciones del gradiente de computación en los ejes x e y .

Un estudio de diferentes métodos de detección de bordes se puede encontrar en (Ziou y Tabbone 1998)¹³. Detección de bordes en la Enciclopedia de las matemáticas [2.5.3] y Enciclopedia de la Ciencia de la Computación e Ingeniería.¹⁴

¹³D. Ziou and S. Tabbone (1998) "Edge detection techniques: An overview", International Journal of Pattern Recognition and Image Analysis, 8(4):537–559, 1998.

¹⁴<http://mrw.interscience.wiley.com/emrw/9780470050118/ecse/article/ecse603/current/abstract>

Capítulo 5

Binarización

5.1. Imagen binaria

5.1.1. Introducción

Una imagen binaria es una imagen digital que tiene sólo dos valores posibles para cada píxel. Típicamente, los dos colores utilizados para una imagen binaria son de blanco y negro. El color usado para el objeto de la imagen es el color del primer plano, mientras que el resto de la imagen es el color de fondo.

Las imágenes binarias son también conocidas como de doble nivel o dos niveles. Esto significa que cada píxel se almacena en un único bit (0 o 1). Nombres como negro-blanco, B&W, blanco y negro, monocromático, son usado para denotar este concepto, pero también suele identificar a la imagen que tiene sólo una muestra por píxel, como escala de grises.

[8]

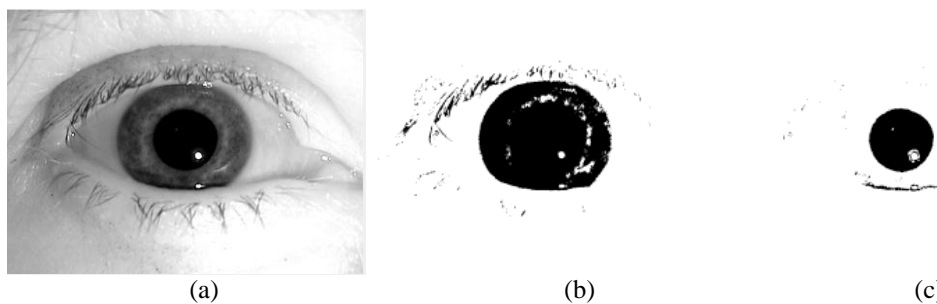


Figura 5.1.1: (a) imagen original. (b) Umbralización (Umbral: 95. Tiempo de ejecución: 0.022 segundos). (c) Umbralización (Umbral: 25. Tiempo de ejecución: 0.011 segundos).

5.1.2. Resultados

Tratando de evitar que se haga muy pesada la lectura, sólo vamos a mostrar ejemplos más representativos de los resultados obtenidos. Sobre ellos es en los que nos hemos basado para llegar a la conclusión final.

Vemos a continuación distintas imágenes con diferentes valores para el filtro binario, así podemos ver cómo afecta a la imagen resultante

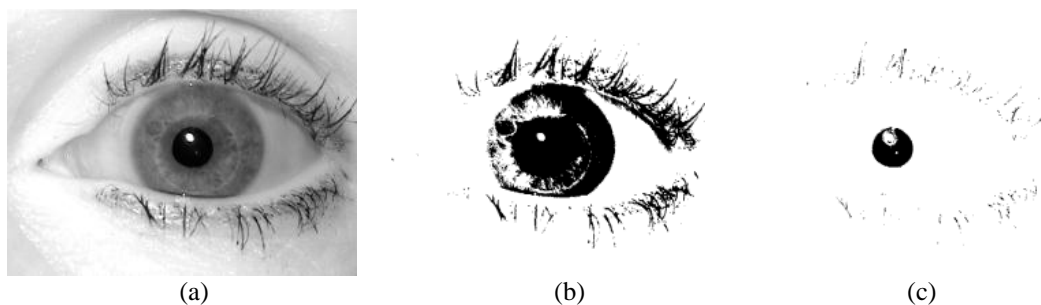


Figura 5.1.2: 287780_244901_I.bmp. a) imagen original. (b) Umbralización (Umbral: 95. Tiempo de ejecución: 0.022 segundos). (c) Umbralización (Umbral: 20. Tiempo de ejecución: 0.015 segundos).

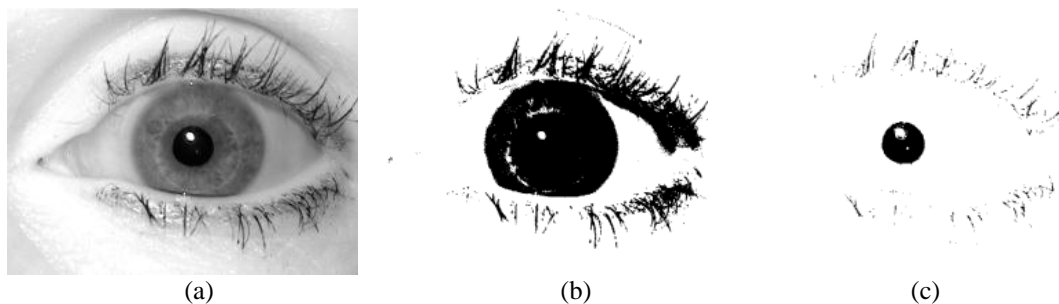


Figura 5.1.3: 287780_244901_I.bmp, imagen. a) imagen original. (b) Umbralización (Umbral: 130. Tiempo de ejecución: 0.018 segundos). (c) Umbralización (Umbral: 25. Tiempo de ejecución: 0.008 segundos).

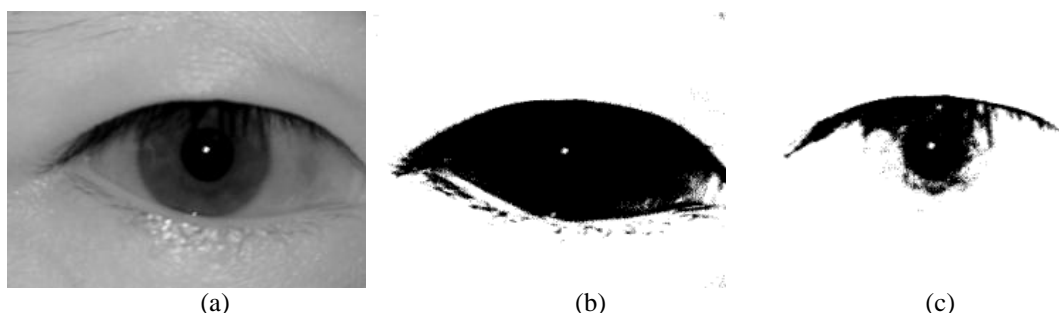


Figura 5.1.4: 289832_246366_D.bmp. a) imagen original. (b) Umbralización (Umbral: 130. Tiempo de ejecución: 0.018 segundos). (c) Umbralización (Umbral: 35. Tiempo de ejecución: 0.008 segundos).

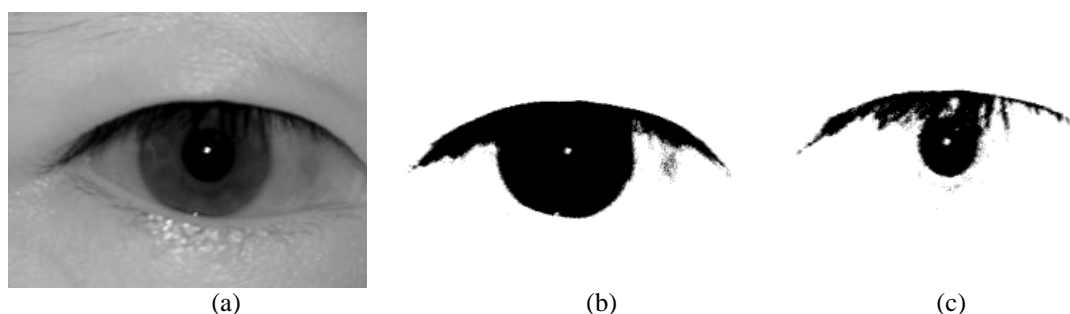


Figura 5.1.5: 289832_246366_D.bmp. a) imagen original. (b) Umbralización (Umbral: 90. Tiempo de ejecución: 0.022 segundos). (c) Umbralización (Umbral: 30. Tiempo de ejecución: 0.015 segundos).

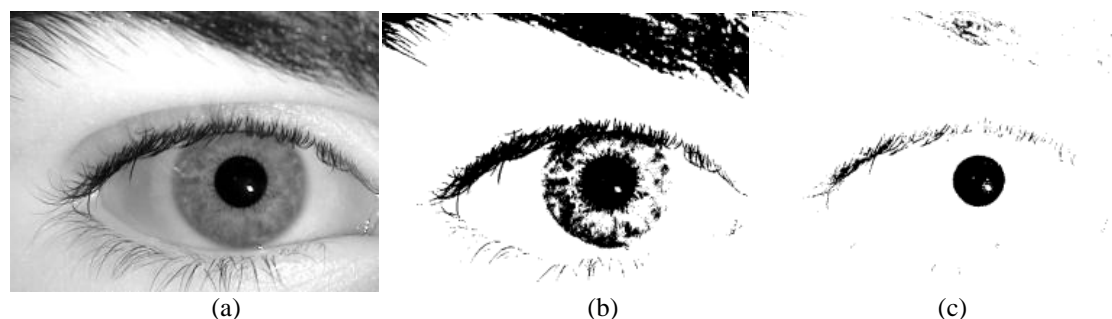


Figura 5.1.6: 290821_240036_D.bmp. a) imagen original. (b) Umbralización (Umbral: 110. Tiempo de ejecución: 0.020 segundos). (c) Umbralización (Umbral: 25. Tiempo de ejecución: 0.013 segundos).

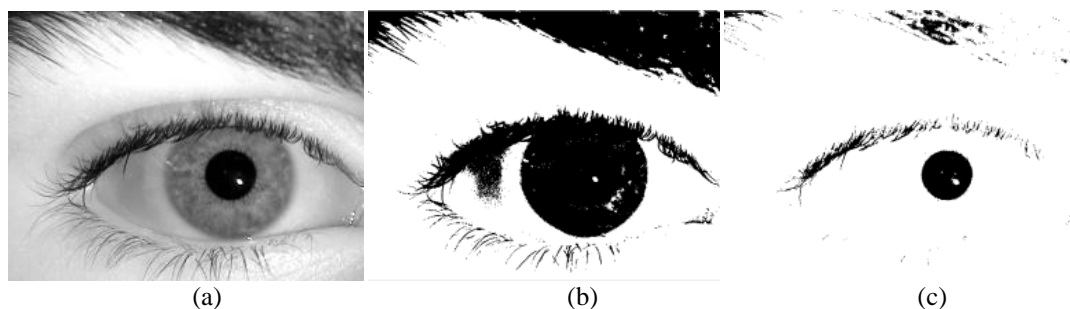


Figura 5.1.7: 290821_240036_D.bmp. a) imagen original. (b) Umbralización (Umbral: 140. Tiempo de ejecución: 0.017 segundos). (c) Umbralización (Umbral: 30. Tiempo de ejecución: 0.008 segundos).

Debido a la gran variedad de las imágenes de entrada, se ha decidido realizar un estudio de diferentes umbrales. Para ello se han considerado un total de 500 imágenes elegidas aleatoriamente de la base de datos. La forma más común de evaluar imágenes binarias es mediante la utilización del valor umbral de una imagen a escala de grises; es decir se elige un valor límite (o bien un intervalo) a partir del cual todos los valores de intensidades mayores serán codificados como 1 mientras que los que estén por debajo serán codificados a cero. Considerando el borde interno se han evaluado los umbrales de 20, 30, 40, y 50. Para el borde externo se han evaluado los umbrales desde 60 hasta 130, ambos inclusive.

		Borde interno			
Umbral		20	30	40	50
Borde	Adecuado	96%	93%	87%	77%
	Posibles errores	3%	5%	9%	15%
	Malo	1%	2%	4%	8%

Tabla 5.1.2.1: Análisis I de los resultados.

		Borde externo			
Umbral		130	120	110	100
Borde	Adecuado	65%	71%	57%	25%
	Posibles errores	32%	24%	28%	55%
	Malo	3%	5%	15%	20%

Tabla 5.1.2.2: Análisis II de los resultados.

		Borde externo			
Umbral		90	80	70	60
Borde	Adecuado	2%	0,5%	0,2%	0%
	Posibles errores	7%	4,5%	1,8%	0,5%
	Malo	91%	95%	98%	99,5%

Tabla 5.1.2.3: Análisis III de los resultados.

	Tiempo de ejecución
Media	0.017833 segundos

Tabla 5.1.2.4: Tiempos de ejecución.

5.1.3. Conclusión

Los resultados logrados son satisfactorios ya que gracias al estudio realizado sobre los umbrales de binarización se ha conseguido ajustar el método eficazmente a las diferentes imágenes. Como se puede observar en los resultados mostrados anteriormente, conseguir resultados satisfactorios está relacionado directamente con el umbral dado.

Los resultados para el borde interno son muy satisfactorios, ya que logramos que se ajuste correctamente en el mayor número de casos con el valor de 20 para el umbral. Al aumentar dicho valor observamos una disminución de las imágenes umbralizadas con bordes adecuados para su posterior uso. Esta disminución llega a reducirse a un 77% con un umbral de valor 50.

Para el borde exterior, se ha realizado un experimento similar para probar diferentes umbrales con el objetivo de conseguir que uno de ellos se adapte correctamente. Sin embargo, aunque se consiguen buenos resultados, consideramos que se pueden conseguir unos mejores. El éxito del algoritmo para el borde exterior, al igual que en el caso anterior, viene relacionado directamente con el umbral. Dependiendo el tipo de ojo se ajustará correctamente un umbral u otro.

Para ojos claros o asiáticos, normalmente se necesita un umbral inferior para conseguir buenos resultados. Sin embargo, para ojos europeos y oscuros, o asiáticos oscuros, el umbral es mayor. Esto es debido a que los diferentes tonos de grises necesiten distintos umbrales para conseguir una imagen binaria aceptable.

Por estos motivos se estudió el uso de un método (descrito a continuación) de ajuste del umbral automático, lo podemos ver en el siguiente punto.

5.2. Método de Otsu

El método de Otsu se utiliza normalmente para realizar automáticamente el histograma de una imagen basado en su umbral, es decir, reducir desde una imagen de grises en una imagen binaria. El algoritmo asume que la imagen umbralizada contiene dos clases de píxeles, histograma bimodal, por lo que calcula el umbral óptimo separando éstas dos clases.

5.2.1. Método

El método de Otsu (1979)¹⁵¹⁶ busca de forma exhaustiva el umbral que minimiza la varianza intraclase, que se define como una suma ponderada de las varianzas de las dos clases:

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

Pesos ω_i son las probabilidades de las dos clases separadas por un umbral t y las variaciones de estas clases σ_i^2 . Otsu muestra que minimizando la varianza intraclase es lo mismo que maximizar la varianza interclase:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]$$

se expresa en términos de probabilidades ω_i y las medias μ_i .

La probabilidad $\omega_1(t)$ se calcula a partir del histograma como t :

$$\omega_1(t) = \sum_0^t p(i)$$

Mientras que la media clase $\mu_1(t)$ es:

$$\mu_1(t) = \sum_0^t p(i)x(i)$$

donde $x(i)$ es el valor en el centro de la bandeja de histograma. Del mismo modo, se puede calcular $\omega_2(t)$ y μ en el lado derecho del histograma.

Las probabilidades así como las medias de clase se pueden calcular de forma iterativa. Esta idea permite que se obtenga un algoritmo efectivo.

5.2.2. Algoritmo

Los pasos a seguir en este método son los siguientes:

1. Calcular el histograma y las probabilidades de cada nivel de intensidad.
2. Establecer los valores iniciales de $\omega_i(0)$ y $\mu_i(0)$.
3. Paso a través de todos los posibles umbrales $t = 1..$ de máxima intensidad posible.
 1. Actualizar $\omega_i()$ y $\mu_i()$.
 2. Calcular $\sigma_b^2(t)$.
4. Umbral deseado corresponde a maximizar $\sigma_b^2(t)$.

¹⁵Lecture notes on thresholding <http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf>

¹⁶Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

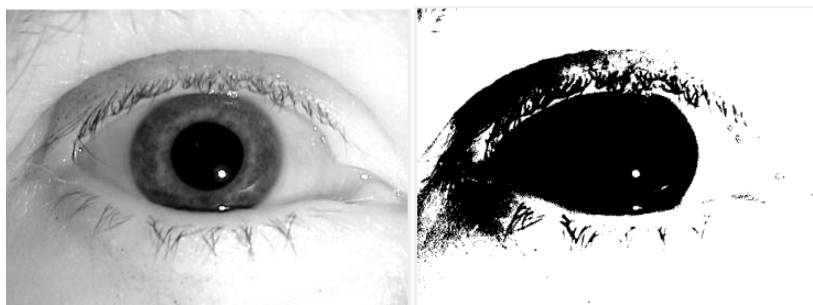


Figura 5.2.1: imagen a la izquierda, original. Imagen a la derecha después de aplicar el método de Otsu. Umbral óptimo 158. Tiempo de ejecución: 0.0501 segundos).

El método de *Otsu* también puede ser extendido a un umbral multinivel, el cual podría dar lugar a una buena segmentación, esta modificación se presentó en el trabajo.¹⁷

5.2.3. Resultados

Hemos evaluado el método de Otsu con total de 500 imágenes de la base de datos utilizada en este trabajo. Tratando de no hacer pesada la lectura, de todas las imágenes analizadas se han mostrado los resultados que se han considerado más significativos.

En el apartado de binarización de imágenes anterior, el ajuste se realizaba de forma manual. En este caso, es el propio método el que ajusta automáticamente.

A continuación, mostramos algunos ejemplos de las imágenes obtenidas aplicando dicho método sobre la base de datos de prueba:

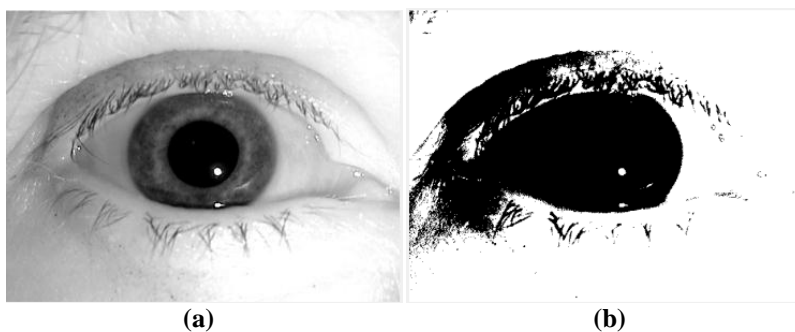


Figura 5.2.2: 287269_241852_D.bmp. (a) Imagen original. (b) Imagen después de aplicar el método de Otsu (Umbral óptimo 158. Tiempo de ejecución: 0.0501 segundos).

¹⁷Ping-Sung Liao, Tse-sheng chenand, Pau-Choo Chung “A Fast Algorithm for Multilevel Thresholding”. Journal of information of science and engineering 17, 713-727. 2001<http://www.iis.sinica.edu.tw/page/jise/2001/200109_01.pdf>

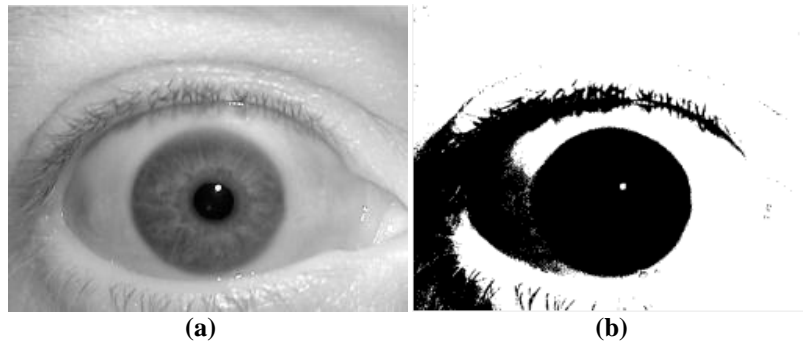


Figura 5.2.3: 287810_243007_D.bmp. (a) Imagen original. (b) Imagen después de aplicar el método de Otsu (Umbral óptimo 152. Tiempo de ejecución: 0.0462 segundos).

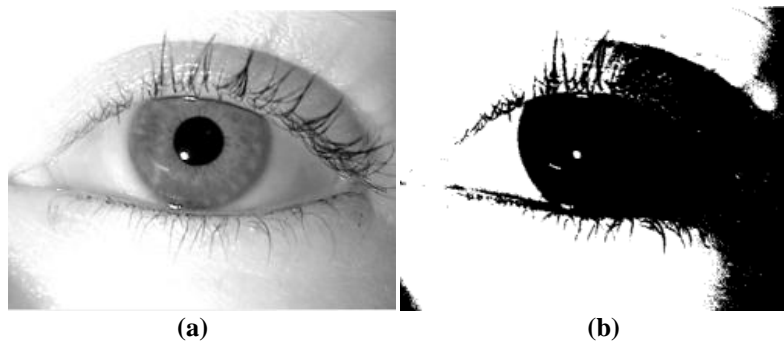


Figura 5.2.4: 28288615_245178_I.bmp. (a) Imagen original. (b) Imagen después de aplicar el método de Otsu (Umbral óptimo 177. Tiempo de ejecución: 0.0453 segundos).

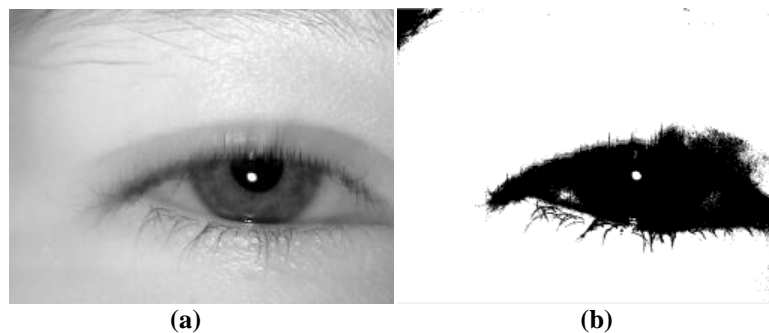


Figura 5.2.5: 288615_245453_D.bmp. (a) Imagen original. (b) Imagen después de aplicar el método de Otsu (Umbral óptimo 148. Tiempo de ejecución: 0.0208 segundos).

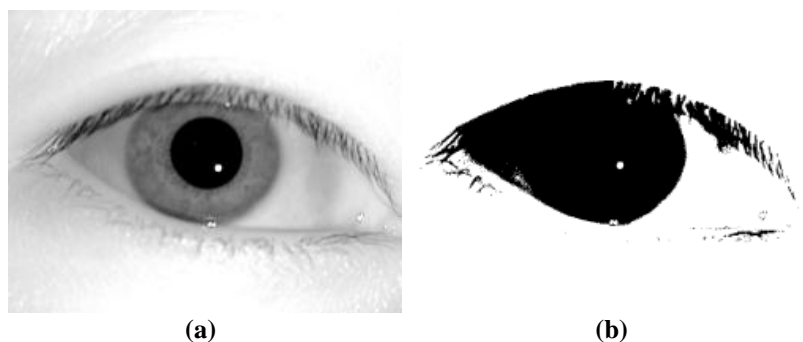


Figura 5.2.6: 288766_241372_D.bmp. (a) Imagen original. (b) Imagen después de aplicar el método de Otsu (Umbral óptimo 161. Tiempo de ejecución: 0.0502 segundos).

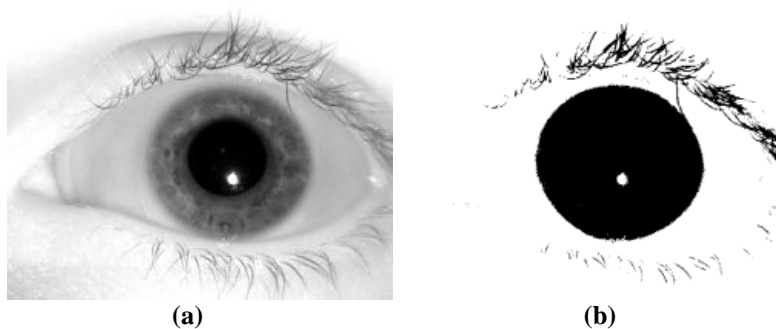


Figura 5.2.7: 290880_245443_I.bmp. (a) Imagen original. (b) Imagen después de aplicar el método de Otsu (Umbral óptimo 150. Tiempo de ejecución: 0.0492 segundos).

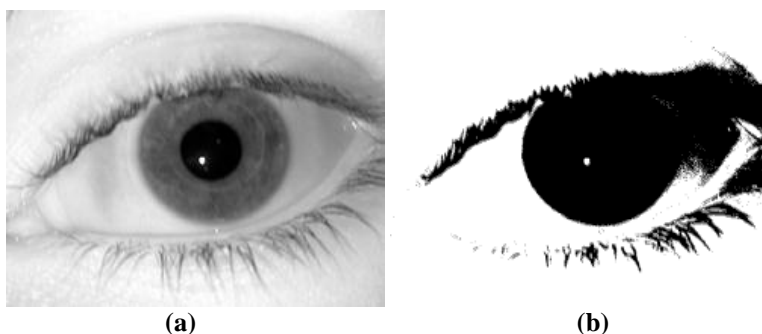


Figura 5.2.8: 290882_244437_I.bmp. (a) Imagen original. (b) Imagen después de aplicar el método de Otsu (Umbral óptimo 158. Tiempo de ejecución: 0.05 segundos).

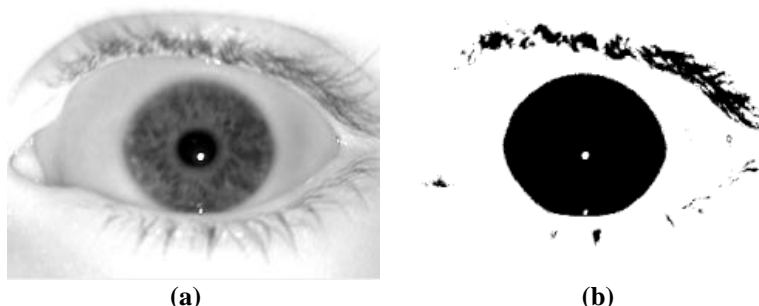


Figura 5.2.9: 291008_242007_I.bmp. (a) Imagen original. (b) Imagen después de aplicar el método de Otsu (Umbral óptimo 155. Tiempo de ejecución: 0.0441 segundos).

5.2.4. Conclusión

Los resultados logrados no son considerados satisfactorios. Aunque se ha conseguido ajustar el método eficazmente a cada una de las imágenes no se obtienen resultados satisfactorios porque como se puede observar en los resultados mostrados anteriormente, no es el umbral que estamos buscando para detectar los bordes del iris.

En los siguientes ejemplos podemos ver, como se ajusta correctamente en imágenes en las que el contraste de tonalidades y colores son mayores, que evidentemente no es nuestro caso. Los tiempos de ejecución en estos casos también pueden considerarse como satisfactorios.



Figura 5.2.10: 291254_242677_D.bmp. (a) Imagen original. (b) Imagen despu3 de aplicar el m3todo de Otsu (Umbral 3ptimo 120. Tiempo de ejecuci3n: 0.0440 segundos).



Figura 5.2.11: Shrikrishna.bmp. (a) Imagen original. (b) Imagen despu3 de aplicar el m3todo de Otsu (Umbral 3ptimo 135. Tiempo de ejecuci3n: 0.0392 segundos).

La Tabla 5.2 muestra un resumen de los resultados obtenidos:

	Umbral de Otsu	Tiempo de ejecuci3n
Media	141.3529	0.045523529segundos

Tabla 5.2.1: An3lisis II resultados m3todo de Otsu.

El umbral resultante es algo superior a 128 que denota la mitad entre 0 y 255. Se puede ver que cuantos m3s niveles de grises tenga la imagen, mejor se ajustar3 este algoritmo. En las im3genes de color se consigue ajustar bien el algoritmo ya que en estas im3genes la diferencia de grises tiende a tener m3s niveles de grises, sin embargo en las im3genes de nuestra base de datos, al tener poco contraste de colores, la diferencia de grises tiene menos niveles. Adem3s el contraste entre los colores oscuros y los colores claros, permite que se ajusta mejor el algoritmo, que tampoco se da en nuestro caso.

Capítulo 6

Algoritmo de *Canny*

El detector de bordes de *Canny* es un modelo que utiliza un algoritmo de múltiples etapas para detectar una amplia gama de bordes en las imágenes. Fue desarrollado por *John F. Canny* en 1986^{18 19}. *Canny* también produjo una teoría computacional de la detección de bordes que explica por qué la técnica funciona.

A pesar de que es muy antiguo, se ha convertido en uno de los estándares de detección de bordes, y aún se sigue utilizando en la investigación.^{20 21}

6.1. Desarrollo del algoritmo de *Canny*

El objetivo de *Canny* era descubrir el algoritmo de detección de bordes óptimo. En esta situación, un detector de bordes óptimo debe cumplir:

- Una buena detección: el algoritmo debe marcar tantos bordes reales de la imagen como sea posible. La probabilidad de detectar los puntos reales de la imagen se debe maximizar, mientras que la probabilidad de falsa detección se debe minimizar, es decir, buscar la maximización de la señal-ruido.
- Buena localización: los bordes marcados deben estar lo más cerca posible del borde de la imagen real.
- Mínima respuesta: una característica en la imagen sólo se debe marcar una vez y siempre que sea posible, el ruido de la imagen no debe crear falsos bordes. (Está implícitamente incluido en el primer requisito)

Para satisfacer estos requisitos *Canny* utiliza el cálculo de variaciones, una técnica que encuentra la función que optimiza una función $f(x)$ dada. La función óptima definida por *Canny* está descrita por la suma de cuatro funciones exponenciales, pero se puede aproximar por la primera derivada de una función *gaussiana*.

6.2. Etapas del algoritmo de *Canny*

El algoritmo se divide en 5 etapas:

1. Suavizado: eliminar el ruido difuminándola imagen. Implementado con un filtro *gaussiano* con tamaño de núcleo específico (N) y el parámetro *sigma*.
2. Encontrar gradientes: los bordes deben estar marcados donde el gradiente de la imagen tenga grandes valores de intensidad.
3. Suprimir los puntos que no sean máximos: sólo los máximos locales se deben marcar como bordes. Encontramos la dirección del gradiente y usamos esta dirección para eliminar los puntos que no sean máximos.
4. Umbral doble: los bordes potenciales vienen determinados por estos umbrales.

¹⁸ John Canny official Berkeley web <http://www.cs.berkeley.edu/~jfc/>

¹⁹ John Canny. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698, Nov. 1986.

²⁰ F. Mai, Y. Hung, H. Zhong, and W. Sze. A hierarchical approach for fast and robust ellipse extraction. Pattern Recognition, 41(8):2512–2524, August 2008.

²¹ Sergei Azernikov. Sweeping solids on manifolds. In Symposium on Solid and Physical Modeling, pages 249–255, 2008.

5. Seguimiento del borde mediante hist3resis: los bordes finales se determinan eliminando los bordes que no est3n conectados a los borde fuertes. Esto se realiza mediante una funci3n recursiva que realiza un umbral doble sobre ambos umbrales: el alto y el bajo.



Figura 6.1: imagen usada como ejemplo de la detecci3n de bordes mediante Canny.

Veamos m3s detalladamente cada una de las etapas de este algoritmo:

1. Eliminar el ruido

Es inevitable que todas las im3genes tomadas por una c3mara tengan una cierta cantidad de ruido presente. Para evitar que el ruido se confunda con bordes en la imagen, el ruido debe ser eliminado.

El detector de bordes de *Canny* utiliza un filtro basado en la primera derivada de una gaussiana. Para empezar, la imagen se convoluciona con un filtro *Gaussiano*. El resultado es una versi3n ligeramente borrosa de la imagen original que no se ve afectada de manera significativa por pixeles ruidoso.

El siguiente es un ejemplo de un filtro *Gaussiano* 5x5, con $\sigma = 1.4$:

$$B = \frac{1}{59} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

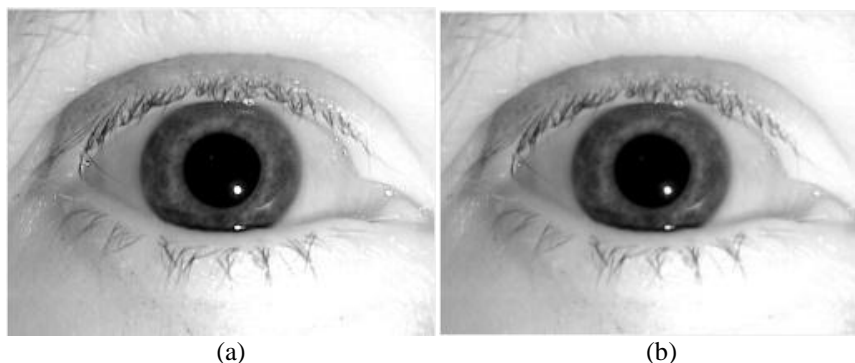


Figura 6.2: (a) Imagen original. (b) Imagen ejemplo despu3s de aplicar el filtro Gaussiano.

2. Encontrar gradiente de alta intensidad en la imagen

El algoritmo de *Canny* localiza bordes donde la variación de intensidad en la imagen es mayor. Estas áreas son localizadas determinando el gradiente de la imagen.

Un borde de una imagen puede apuntar a varias direcciones, de modo que el algoritmo de *Canny* utiliza cuatro filtros para detectar los bordes horizontales, verticales y diagonales en la imagen borrosa. El operador de detección de bordes (*Roberts*, *Prewitt*, *Sobel*, por ejemplo) devuelve un valor para la primera derivada en la dirección horizontal (G_x) y la dirección vertical (G_y). Dicho de otra manera, calcula mediante el operador de *Sobel*, los gradientes en cada píxel de la imagen. El primer paso es aproximar los gradientes en las direcciones horizontal y vertical respectivamente, aplicando los siguientes operadores:

Sobel:

$$K_{GX} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}; K_{GY} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Prewitt:

$$K_{GX} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}; K_{GY} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

Las magnitudes del gradiente, que son los puntos fuertes de los bordes, se pueden determinar mediante la distancia Elucídela y aplicando el teorema de Pitágoras:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

En ocasiones puede ser simplificado utilizando la distancia de *Manhattan*, para reducir la complejidad computacional.

$$|G| = |G_x| + |G_y|,$$

en ambos casos, G_x y G_y , son gradiente en las dirección x e y respectivamente.

La magnitud de los gradientes normalmente indican los bordes con bastante claridad, pero típicamente, los bordes son gruesos, por lo que no se sabe dónde está el borde exactamente. Para saber dónde está exactamente los bordes se pueden calcular mediante la ecuación:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right).$$

El ángulo de dirección del borde se aproxima a una de cuatro ángulos: vertical, horizontal y los dos diagonales (0, 45, 90 y 135 grados, por ejemplo). [3.1.1]

3. Supresión de puntos que no sean máximos

El propósito de esta etapa es convertir los bordes borrosos de la imagen, en bordes finos. Básicamente se hace quedándonos con todos los máximos locales de la imagen, y borrando todo lo demás. El algoritmo calcula el gradiente de la imagen para cada píxel.

Cuando la dirección del gradiente es cercana a los 45°:

- A. Se compara la resistencia del borde del píxel actual con la resistencia del borde del píxel en la dirección dada, y en la contraria. Esto es, si la dirección es 90°, comparar con el Norte y con el Sur.
- B. Si la resistencia del borde del píxel actual es mayor, nos quedamos con este valor, si no es así, eliminamos el valor.

Un ejemplo sencillo de la supresión de puntos que no sean máximos:

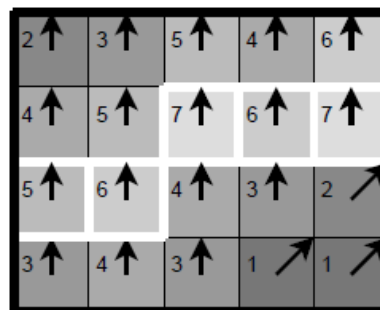


Figura 6.3: ilustración que muestra la supresión de píxeles que no son máximos. La fuerza de los bordes está indicada en números, mientras las flechas muestran la dirección del gradiente. El borde resultante son los píxeles marcados con borde blanco.

En las imágenes que nos conciernen, la mayoría de los píxeles apuntan hacia el norte, por lo que todos son comparados con los de por encima y por debajo. Los píxeles que resultan ser máximos, se marcan con un recuadro de borde blanco. El resto de píxeles se suprime.

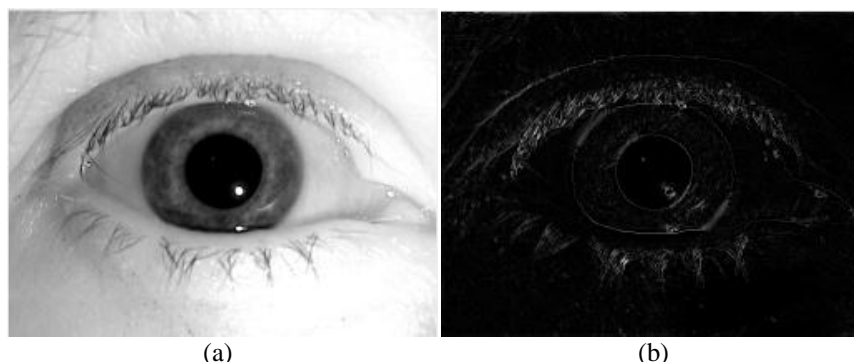


Figura 6.4: (a) Imagen original. (b) Imagen ejemplo después de eliminar los puntos que no son máximos.

4. Doble umbral

Los bordes que quedan tras la etapa de supresión de puntos que no son máximos no muestran la fuerza de cada píxel. Muchos de ellos serán bordes reales, pero

algunos pueden ser causados por las variaciones de color o ruido en las superficies rugosas. La forma más sencilla de solucionar esto sería mediante un umbral, de modo que sólo los bordes que superen ese umbral se conservaran. El algoritmo de detección de bordes Canny utiliza un umbral doble. Los bordes que superen el umbral alto, serán clasificados como fuertes, los que por debajo del umbral bajo se eliminan, y los que están entre ambos umbrales se clasifican como débiles.

Umbralización con histéresis requiere de dos umbrales, alto y bajo. Haciendo la suposición de que los bordes importantes deben ser curvas continuas en la imagen, esto nos permite seguir una línea débil y poder descartar algunos píxeles ruidosos que no constituyen una línea, pero sí han producido niveles grandes de intensidad en los gradientes. Comenzamos la umbralización mediante la aplicación de un umbral alto. Esto marca los bordes que podemos estar seguros son reales. A partir de estos, utilizando la información derivada de la dirección anterior, todos los bordes pueden ser localizados en la imagen. Mientras que para el seguimiento de un borde, se aplica el umbral más bajo, lo que nos permite localizar las secciones débiles de los bordes, siempre y cuando nos encontramos con un punto de partida, es decir, un borde fuerte adyacente.

Una vez completado este proceso tenemos una imagen binaria en la que cada pixel quedó marcado, ya sea como un píxel que represente un borde o un píxel que no lo represente.

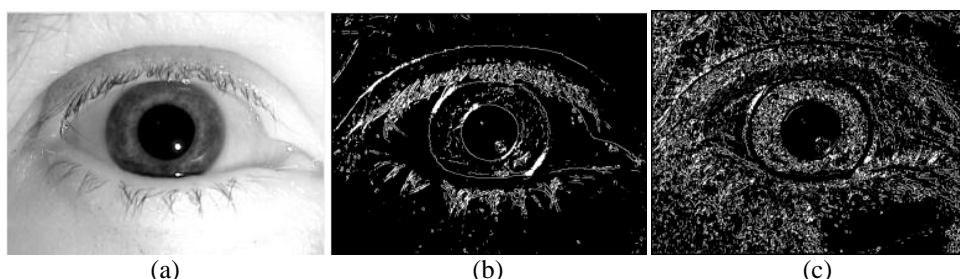


Figura 6.5:(a) Imagen original. Imagen después de aplicar el doble umbral. (a) bordes fuertes (b) bordes débiles.

5. Seguimiento del borde por histéresis

Los bordes fuertes son interpretados como bordes reales e inmediatamente se pueden incluir en la imagen final. Los bordes débiles se incluyen si y sólo si están conectados a un borde fuerte. Esto se debe a que el ruido y otras pequeñas variaciones no tienen como resultado un borde fuerte (asumiendo que se ha ajustado los umbrales adecuadamente). Así los bordes fuertes, en la gran mayoría, sólo podrán ser bordes reales de la imagen original. Los bordes débiles pueden ser debidos a bordes reales, o a ruido, o variaciones de color. Los bordes débiles debido a los bordes reales son mucho más propensos a estar directamente conectados a los bordes fuertes. Sin embargo, los bordes débiles que vienen de variaciones de color estarán distribuidos de forma independiente a los bordes fuertes. [9]

6.3. Parámetros

El algoritmo de Canny contiene un número de parámetros ajustables, que pueden afectar en el tiempo de cálculo y en la eficacia del algoritmo:

- El tamaño del filtro Gaussiano: el filtro de suavizado utilizado en la primera etapa directamente afecta a los resultados del algoritmo de Canny. Filtros más pequeños causan menos borrosidad y permitir la detectar líneas más pequeñas. Un filtro mayor causa más borrosidad, por lo que ciertos bordes quedaran eliminados, o pasarán a considerarse como ruido o bordes débiles. Aumentar la difuminación, es decir con un filtro más grande, es más útil para la detección de bordes grandes.
- Umbrales: el uso de dos umbrales con histéresis permite una mayor flexibilidad que en un enfoque de un solo umbral, pero los problemas de la doble umbralización siguen presentes. Un umbral demasiado alto puede perder información importante. Por otro lado, un umbral demasiado bajo identificará información irrelevante (tales como el ruido) como información importante. Es difícil definir un umbral genérico que funcione bien en todas las imágenes.[8]

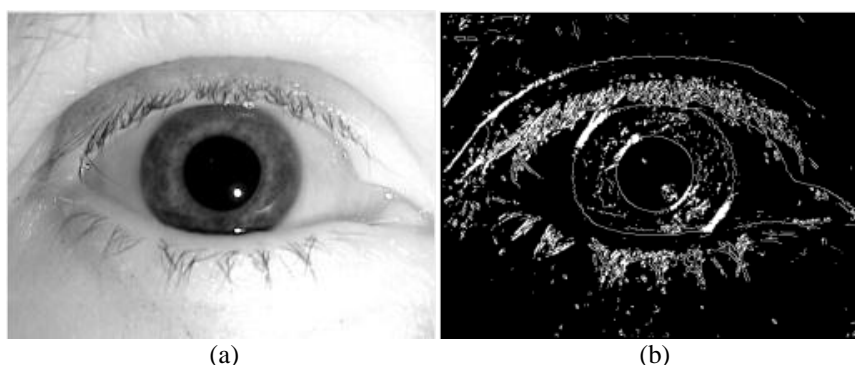


Figura 6.6: (a) imagen original. (b) imagen después de aplicar el algoritmo Canny.

Sigma	9
Tamaño de la máscara del filtro Gaussiano	1
Umbral alto	30
Umbral bajo	10
Tiempo de ejecución	1.46 seg

Tabla 6.1: Parámetros utilizados en el ejemplo del algoritmo de Canny para la figura 5.6

6.4. Resultados Canny

Tras desechar los dos métodos anteriores, porque no se ajustaba correctamente a nuestro estudio. El primero de ellos, binarización, aunque se ajustaba correctamente, tenía el inconveniente de que se tenía que ajustar manualmente. El segundo, el método de Otsu, no se ajusta correctamente, ya que en la gran parte de los casos no se ajusta con el

umbral adecuado. Por ello se ha decidido ejecutar *Canny* sin tener en cuenta dichos m3todos.

En los puntos anteriores se han explicado los par3metros ajustables del algoritmo: el tama1o del filtro y los umbrales. El tama1o del filtro, tiene otro par3metro ajustable, sigma. Sigma es la desviaci3n est3ndar de la probabilidad asociada.

En nuestro caso, se ha decido utilizar los siguientes valores:

Par3metros de Canny	
Tama1o del filtro <i>Gaussiano</i>	9

Tabla 6.2: Par3metros utilizados en los resultados del algoritmo de Canny

Se ha elegido este valor del tama1o del filtro *Gaussiano* buscando eliminar posibles reflejos, bordes insignificantes o cualquier caracter3stica que dificulte la detecci3n de los bordes buscados.

Vemos algunos ejemplos de los casos estudiados para tratar de explicar en los que nos hemos basados para llegar a la conclusi3n que hemos llegado.

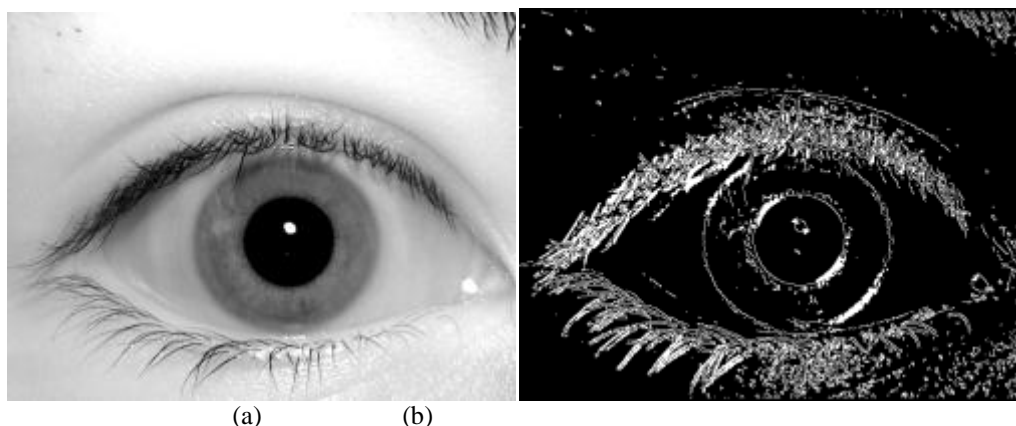


Figura 6.7: 291254_243722_D.bmp, (a) imagen original. (b) imagen despu3s de aplicar el algoritmo de Canny. (Umbrales: 30, 10. Tiempo de ejecuci3n: 2.48299 segundos).

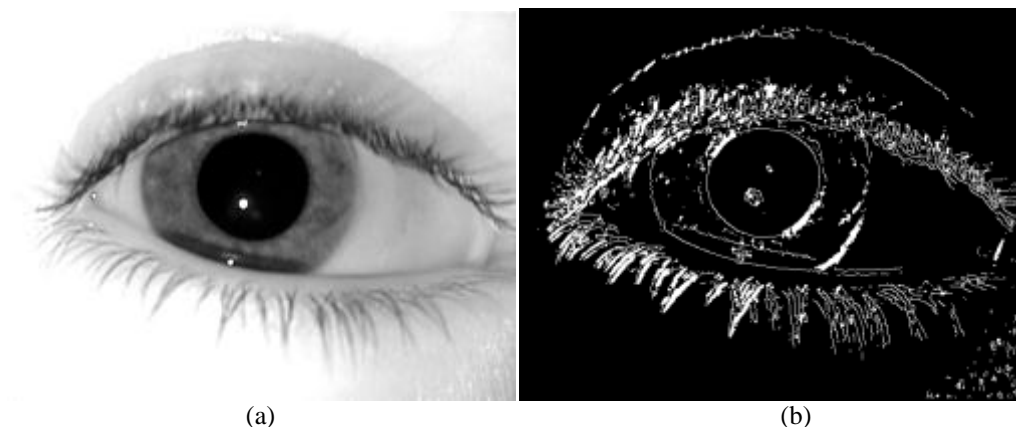


Figura 6.8: 291307_244451_D.bmp, (a) imagen original. (b) imagen despu3s de aplicar el algoritmo de Canny. (Umbrales: 30, 10. Tiempo de ejecuci3n: 2.5275 segundos).

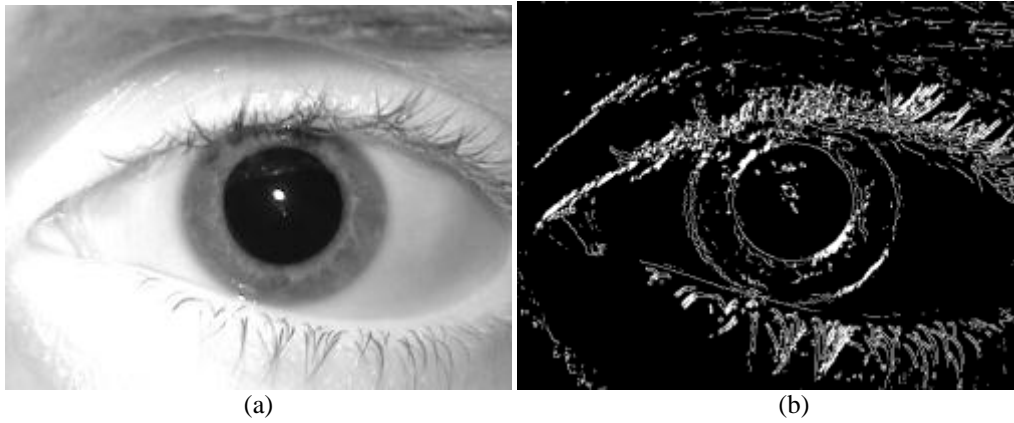


Figura 6.9: 291242_243686_I.bmp, (a) imagen original. (b) imagen después de aplicar el algoritmo de Canny. (Umbrales: 30, 10. Tiempo de ejecución: 2.1699 segundos).

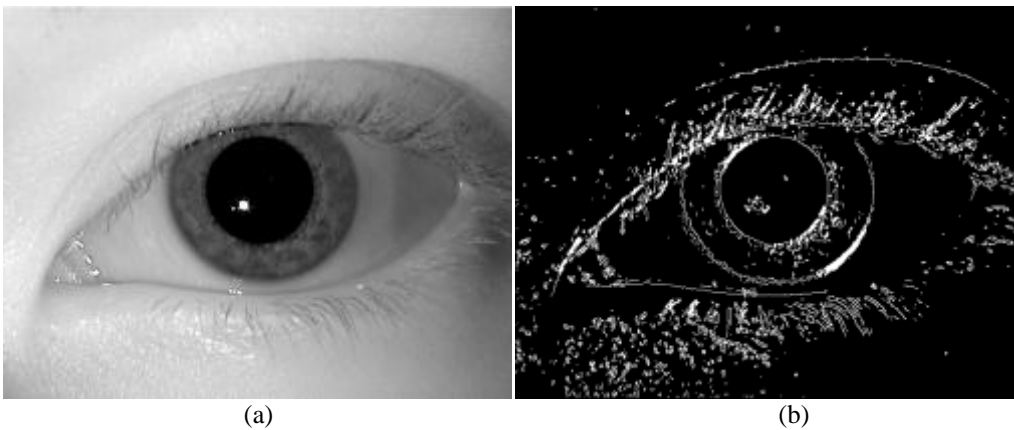


Figura 6.10: 291299_240429_I.bmp, (a) imagen original. (b) imagen después de aplicar el algoritmo de Canny. (Umbrales: 30, 10. Tiempo de ejecución: 2.2511 segundos).

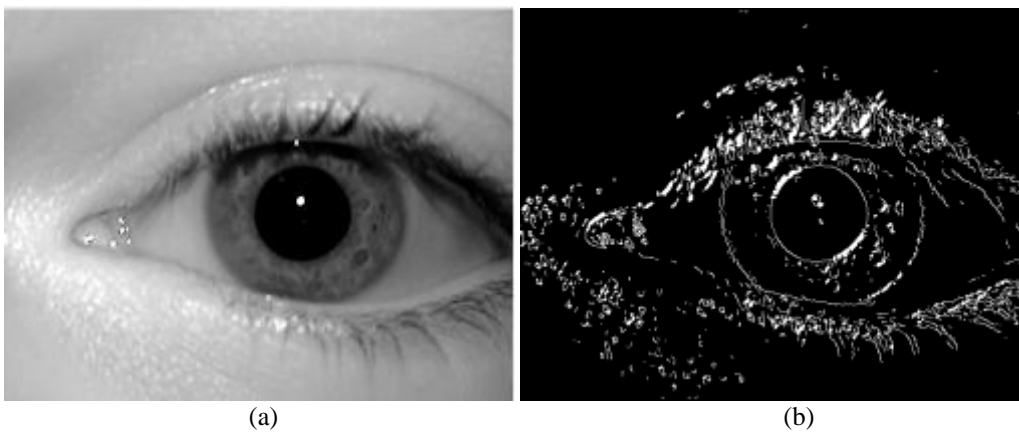


Figura 6.11: 291461_241115_I.D.bmp, (a) imagen original. (b) imagen después de algoritmo de Canny. (Umbrales: 30, 10. Tiempo de ejecución: 2.3273 segundos).

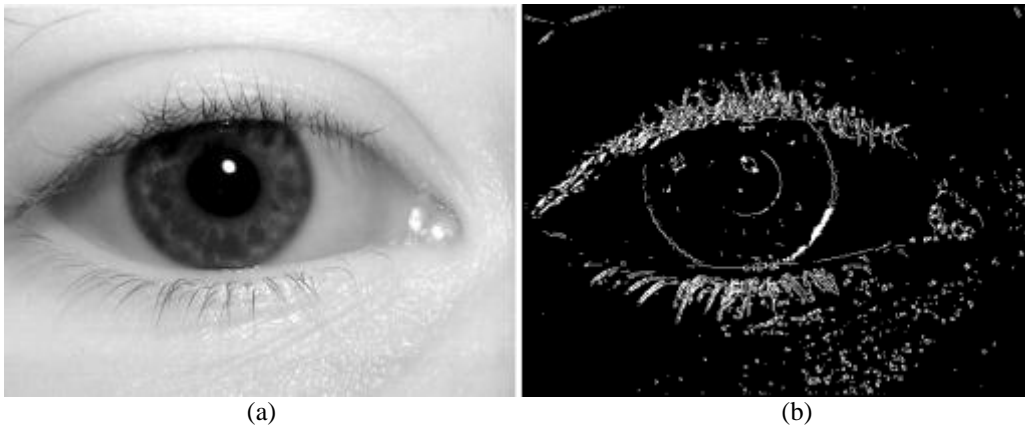


Figura 6.12: 291465_241137_D.bmp, (a) imagen original. (b) imagen después de aplicar el algoritmo de Canny. (Umbrales: 40, 20. Tiempo de ejecución: 2.2949 segundos).

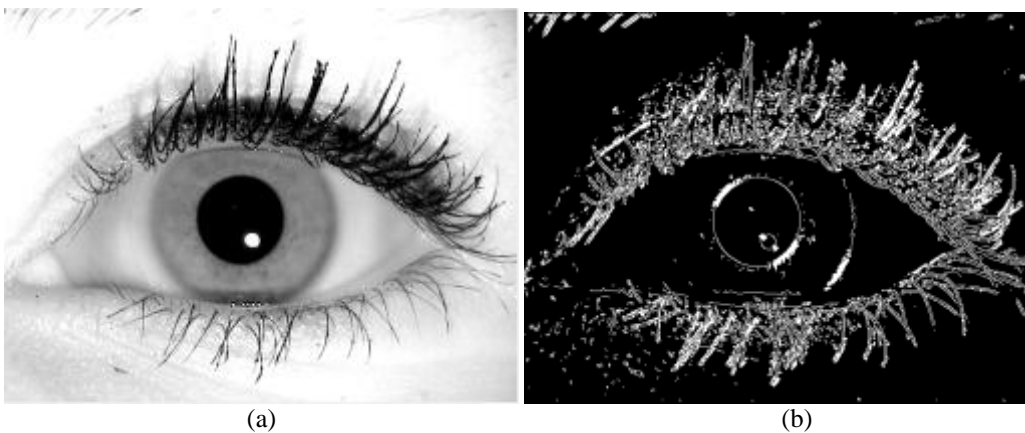


Figura 6.13: 291494_239574_I.bmp, (a) imagen original. (b) imagen después de aplicar el algoritmo de Canny. (Umbrales: 40, 20. Tiempo de ejecución: 2.2964231 segundos).

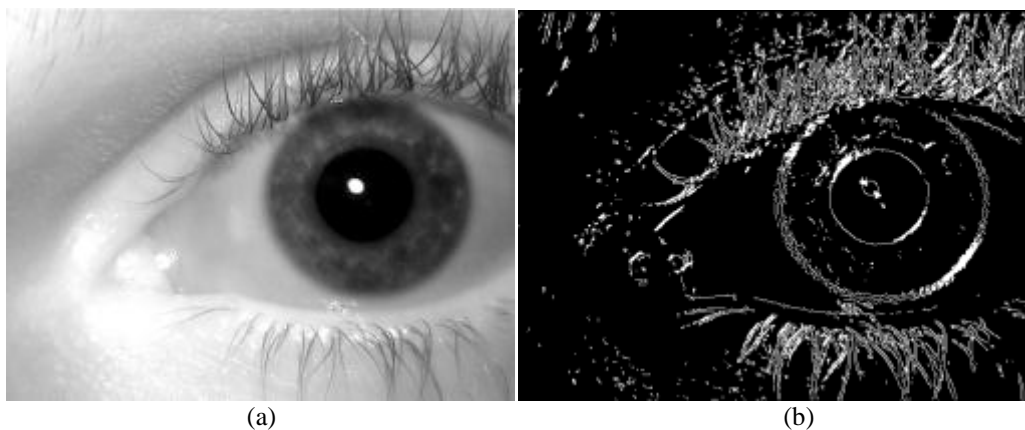


Figura 6.14: 291038_246405_I.bmp, (a) imagen original. (b) imagen después de aplicar el algoritmo de Canny. (Umbrales: 28, 15. Tiempo de ejecución: 2.3653 segundos).

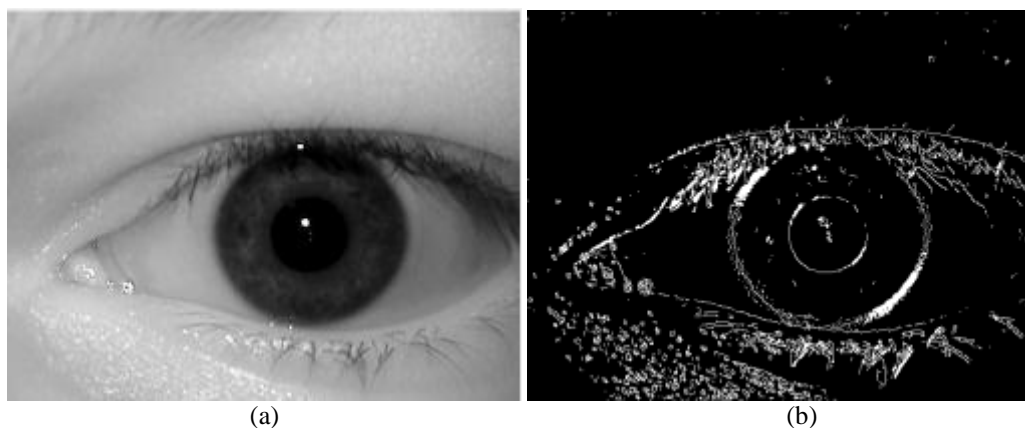


Figura 6.15: 2291238_242648_I.bmp, (a) imagen original. (b) imagen despu3s de aplicar el algoritmo de Canny. (Umbrales: 28, 15. Tiempo de ejecuci3n: 2.06123 segundos).

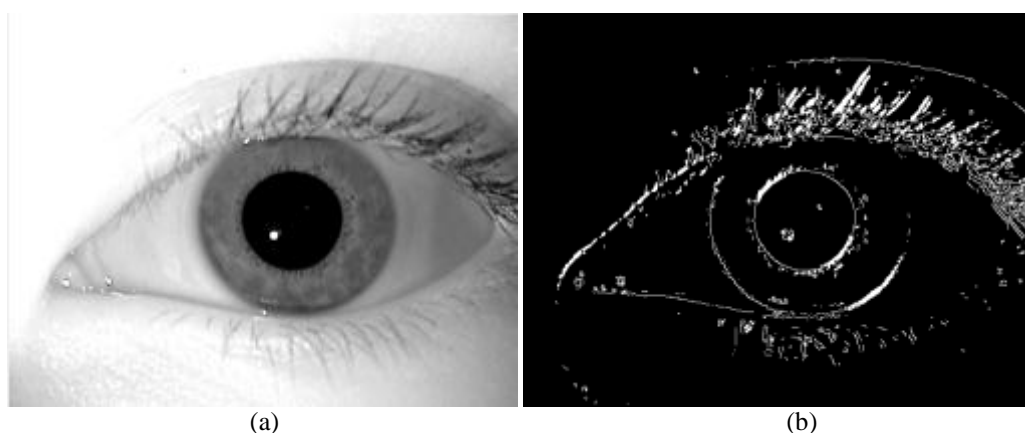


Figura 6.16: 291299_242331_I.bmp, (a) imagen original. (b) imagen despu3s de aplicar el algoritmo de Canny. (Umbrales: 28, 15. Tiempo de ejecuci3n: 2.31165 segundos).

A continuaci3n podemos ver un breve an3lisis de los datos obtenidos:

Tiempo de ejecuci3n	
Media	2,308829

Tabla 6.3: Tiempos de ejecuci3n

Debido a la gran variedad de las im3genes de entrada, se ha decido realizar un estudio de diferentes umbrales. Para ello se han considerado un total de 500 im3genes elegidas aleatoriamente de la base de datos. A continuaci3n vemos una tabla del an3lisis realizado para cada una de las parejas de umbrales.

		Canny						
Umbral alto		25	28	30	30	35	40	50
Umbral bajo		10	15	10	15	20	20	25
Borde	Adecuado	64%	81%	87%	87%	76%	2%	1%
	Posibles errores	25%	13%	8%	8%	13%	11%	1%
	Malo	11%	6%	5%	5%	11%	87%	98%

Tabla 6.4: An3lisis de los resultados despu3s del algoritmo Canny

6.5. Conclusión

El algoritmo de Canny es adaptable a diferentes entornos. Sus parámetros permiten que sea adaptado al reconocimiento de bordes de diferentes características, dependiendo de las necesidades particulares de una aplicación dada. En el texto original de Canny, la derivación del filtro óptimo condujo a un filtro de respuesta finita de impulsos, que puede ser un proceso muy lento en el dominio espacial si hay mucha eliminación de ruido.

Los resultados logrados son plenamente satisfactorios con los parámetros adecuados. Estos parámetros permiten ajustar el algoritmo a distintos tipos de imágenes, especialmente a imágenes con formas circulares. Para nuestro caso, tras probar con diferentes valores para estos parámetros, se ha realizado un estudio para conseguir los valores más adecuados.

Tras observar los resultados de diferentes tipos de imágenes se ha apreciado que entre los umbrales alto y bajo debe haber una relación entre $th:tl = 2:1$ o $3:1$. Es por esta razón por la que se ha decidido seguir este patrón en las pruebas realizadas.

Para los valores de umbrales más altos 50:25, 40:20, los resultados malos se disparan. Se aprecia que no son bordes con la suficiente fuerza como para que sean relevantes en estos umbrales. Para los valores de umbrales más bajos se aprecia como los resultados son mejores, por lo que se ha decido analizar con mayor detalle los valores bajos, hasta llegar a que los valores 30:15 o 30:10 se ajustan correctamente a nuestras imágenes.

Los tiempos de ejecución también se han analizado, y se han considerado satisfactorios.

Capítulo 7

Contornos activos

Dado los problemas con los métodos comunes, las técnicas clásicas de segmentación, que no se puede utilizar una imagen separando los elementos que la constituyen y que no es efectivo en presencia de ruido, artefactos de muestreo, u otros elementos.

Las soluciones posibles a estos inconvenientes es generalizar el uso de plantillas que permitan detectar formas o la *transformada de Hough*, pero las necesidades computacionales y las restricciones en las formas a buscar en la imagen para estos métodos son muy altas, es por ello el deseo de encontrar un método que busque cualquier forma en una imagen que sea suave y que su forma sean contornos cerrados [10].

Estos problemas condicionan la calidad de los resultados aplicando métodos comunes y clásicos, por lo que en ocasiones debemos aplicar un segundo procesado más complejo para alcanzar la calidad requerida en los resultados obtenidos.

Utilizando técnicas más recientes, complejas, y menos comunes solucionamos estos inconvenientes de forma eficiente y satisfactoria. Estas técnicas son más complejas frente a los elementos antes mencionados que distorsionaban el resultado final, es por ello por lo que permiten segmentar imágenes mucho más complejas que las segmentadas con técnicas clásicas, además no requieren de un segundo procesado, y se pueden interpretar, clasificar y reconocer directamente, pero necesitan más tiempo de cálculo. Una de estas técnicas son los contornos activos.

Un contorno activo o contorno deformable puede considerarse como un hilo elástico que envuelve al objeto que se desea identificar. El *snakes*, en realidad, un conjunto de puntos, píxeles, que mediante interpolación forman un contorno. Este contorno, que necesariamente ha de ser inicializado cerca del objeto a identificar y envolviendo al mismo, será capaz de ajustarse a cualquier contorno cerrado. Es decir, a cualquier objeto que se desee reconocer. [11]

Los contornos activos son mecanismos que modelan la síntesis de imágenes, análisis de imágenes, diseño geométrico por computadora y otros campos relacionados. Concretamente ofrecen cierto grado de conocimiento a priori a la interpretación de la imagen.

En general el método de los contornos activos consiste primero en realizar un preprocesado de la imagen para así mejorar la calidad de la imagen y facilitar los procesados posteriores, a continuación según sea la aplicación final del contorno tendremos que eliminar o disminuir ciertos efectos de la imagen, que dificulten la búsqueda del contorno, para así seguir facilitando el procesado final. Para finalizar debemos inicializar el contorno, que es muy importante, ya que una mala inicialización del contorno puede originar un *snake* que no converge a un resultado óptimo.

Los contornos activos los podemos clasificar en *snakes*, contornos dinámicos y superficie deformables. Nos centraremos en los *snakes*, pero también veremos brevemente los contornos dinámicos.

7.1. Introducción

7.1.1. *Snakes*

Los *snakes*, que son un caso especial de los modelos deformables, empezaron a ganar popularidad a partir de los años ochenta, representa la frontera de un objeto u otra propiedad de la imagen que se caracteriza como curva paramétrica. A esta curva se le asocia una energía o fuerza como veremos a continuación. El problema de la búsqueda de la frontera del objeto se presenta como el resultado de minimizar dicha energía, que también veremos a continuación.

Los *snakes* modelan las fronteras entre un objeto, el fondo y el resto de objetos de una imagen y el resto de objetos de la imagen, permiten obtener los contornos de los objetos que utilizan información a priori de la forma de los objetos. Algunas propiedades como la continuidad, curvatura, o la suavidad son impuestas desde el principio, en lugar de esperar que condiciones esperadas de estas propiedades provengan de los datos de imagen.

El objetivo fundamental del *snake* es refinar la solución propuesta por otras técnicas, es decir, si utilizamos otra técnica más clásica obtenemos un contorno cercano a la solución, con un *snake* conseguimos que el contorno se acerca a su mínimo local óptimo, por lo que obtendríamos el contorno deseado. Dicho de otra forma los *snakes* es una técnica de ajuste más eficiente que la del modelo deformable, utilizando la técnica de minimizar la energía.

Otra forma de definir los *snakes*, es como una curva que se encarga de minimizar la energía influenciada por energías como la externa o la energía de la imagen, que suelen encontrarse en características como las líneas y bordes. En las técnicas clásicas, al minimizar la energía pueden resultar varios puntos en los que la energía es mínima, es por eso por lo que elegir la solución correcta es muy importante, en base a un alto razonamiento lógico o la interacción del usuario, sin embargo las nuevas técnicas dan lugar al punto donde la energía queda minimizada eficientemente para su posterior uso. [12]

El problema a resolver está en la localización de bordes, líneas y contornos, además del seguimiento de estos contornos por la imagen. En esta técnica la energía se ve afectada por la presencia de bordes en la imagen y por la conectividad de los contornos, y consecuentemente a la estructura del contorno óptimo. Incluso para mejorarlo nos podremos apoyar en técnicas de alto nivel para llegar hasta el contorno óptimo apropiado. En comparación con otras técnicas de segmentación que se basan en búsquedas de contornos resaltados de la imagen, esto es una técnica activa, que minimiza la función de energía, por lo que va a tener un comportamiento dinámico. [13]

Antes de comenzar con su modelo teórico vamos a diferenciar claramente las energías externas, internas y de imagen.

Las fuerzas externas añaden información de alto nivel para hacer que el *snake* se dirija hacia el mínimo local deseado. La energía de la imagen arrastra al *snake* hacia ciertas características de las imágenes como son las líneas, bordes o contornos subjetivos. Los *snakes*, poseen también energías internas debido a la flexión cuya función es imponer

las restricciones de continuidad, suavidad o curvatura, como comentamos anteriormente, para conseguir los resultados deseados. Esta energía interna se supone que es mínima cuando el *snake* tiene una forma relevante teniendo en cuenta la forma del objeto buscado.

- Energía de contorno: es el término de continuidad. Intenta que el contorno que envuelve al objeto deseado es continuo, y que evite que dicho contorno se estire se encoja.
- Energía de curvatura: es el término de curvatura. Intenta que el contorno se resista a doblarse o retrocederse.

[11] [12]

7.1.2. Modelo teórico tradicional

Geométricamente, el *snake* se representa como una curva paramétrica, variable en el tiempo y definido en el plano de la imagen $(x, y) \in R^2$, mediante $r(s) = ((x(s), y(s)))$, donde las coordenadas x e y del contorno son funciones de las variables paramétricas $s \in (0,1)$. Su funcional de energía se puede poner como:

$$E_{snake}^*(r) = \int_0^1 E_{int}(r(s))ds + \int_0^1 E_{img}(r(s))ds + \int_0^1 E_{res}(r(s))ds,$$

en la que E_{int} representa la energía interna del *snake*, E_{img} la energía o fuerza de la imagen, causada por las características de la imagen, y E_{res} las fuerzas restrictivas externas.

La energía interna del *snake* se puede definir como la suma de la energía de elasticidad y la energía de rigidez:

$$E_{int}(r(s)) = E_{elasticidad} + E_{rigidez}$$

a su vez, cada una de estas energías se pueden definir como veremos a continuación:

$$E_{elasticidad} = \frac{1}{2}(\alpha |r'(s)|^2) = \frac{1}{2}(\alpha \|\frac{d\bar{r}}{ds}(s)\|^2)$$

$$E_{rigidez} = \frac{1}{2}(\beta |r''(s)|^2) = \frac{1}{2}(\beta \|\frac{d^2\bar{r}}{ds^2}(s)\|^2),$$

Quedando la energía interna como la suma de ambas:

$$E_{int}(r(s)) = \frac{1}{2}(\alpha |r'(s)|^2) = \frac{1}{2}(\alpha \|\frac{d\bar{r}}{ds}(s)\|^2) + \frac{1}{2}(\beta |r''(s)|^2) = \frac{1}{2}(\beta \|\frac{d^2\bar{r}}{ds^2}(s)\|^2)$$

La energía interna está formada por un término controlado por α y otro término controlado por β . La derivada de primer orden controla la tensión del contorno, la elasticidad, es decir, el contorno evoluciona lentamente según varía el parámetro s . Sin embargo la derivada de segundo orden controla la rigidez, se comporta como una delgada lámina. Ajustando estos parámetros se puede controlar la importancia relativa de estos términos a lo largo del contorno. [13]

En el caso en el que incrementemos α obligamos a que el *snake* tienda a una parametrización uniforme, incrementando la posibilidad de que encoja y evitando ondulaciones y dobleces, a no ser que se ajusten las fuerzas externas u otros elementos.

En el caso en el que fijamos β a cero para un cierto valor de s , permitimos al *snake* doblarse, es decir, que tenga una discontinuidad de primer orden, o lo que es lo mismo un vértice o esquina. Si en cambio aumentamos el valor de β obligamos a que la curva sea suave, es decir que se aproxime a una línea recta y rígida pero flexible. Estos parámetros los podemos considerar como agentes reguladores del *snake*.

Podemos definir funciones de energía para que atraigan el *snake* hacia las características relevantes de la imagen, es decir, líneas, bordes y terminaciones, cada una de ellas genera una función de energía, y la energía de la imagen es la suma de las tres:

$$I_{image} = w_{line} E_{line} + w_{edge} E_{edge} + w_{term} E_{term},$$

donde los términos w_{line} , w_{edge} , w_{term} , son los pesos correspondientes a la energía de línea E_{line} , la energía de borde E_{edge} , y la de terminación E_{term} , respectivamente. En función de la forma en la que regulemos estos términos podremos modificar el comportamiento del *snake* respecto a la imagen. [14]

Para definir la energía de líneas basta con la intensidad de la imagen, es decir, si una imagen tiene líneas la energía se verá atraída por estas líneas.

$$E_{line} = I(x(s), y(s)),$$

por lo que dependiendo del signo de w_{line} , el *snake* se sentirá atraído hacia las líneas con contornos más claros o más oscuros.

La energía de los bordes de la imagen se puede definir como:

$$E_{edge}(r(s)) = -\|\nabla I(x(s), y(s))\|^2$$

así el *snake* se verá atraído por los gradientes de gran intensidad, y al igual que la energía de línea, según el signo de w_{edge} , será atraído por contornos más claros o más oscuros.

La convergencia del *snake* cuando el contorno atraviere un punto de la imagen de baja energía, como el ruido, el *snake* utilizará la energía de los puntos vecinos para intentar la continuación de la función. Este tipo de convergencia es bastante común en los *snakes*. El hecho de utilizar una energía adecuada, puede originar que se acerque al mínimo local óptimo. Esto puede ocurrir también si suavizamos la energía de las líneas o bordes, es decir, podríamos llegar a un equilibrio de la energía a partir de una imagen borrosa, provocado por ejemplo por la elección de la escala, y una vez que se produzca el equilibrio reducimos las dudas que se generen a partir de la imagen borrosa. [15]

Guiándonos por la teoría de Marr-Hildreth de la detección de bordes²², la energía de línea y bordes sería:

$$E_{line}(r(s)) = -(G * \nabla^2 I(x(s), y(s)))^2$$

$$E_{edge}(r(s)) = -|G_{\sigma}(x(s), y(s)) * \nabla^2 I(x(s), y(s))|^2,$$

de forma que el parámetro G_{σ} es una función gaussiana de desviación típica σ . Los mínimos resultan cuando fijamos a cero la función $(G_{\sigma} * \nabla^2 I)$, que corresponden a los bordes según la teoría de Marr-Hildreth. Añadiendo estos términos de la energía,

²²D.Marr and E. Hildreth, "A theory of edge detection". Proc. Roy. Soc. London B., 207:187-217. 1980

significa que el *snake* se sentirá atraído por los cruces con cero, pero estará limitado por la suavidad de la imagen, es decir, eligiendo una escala σ podremos enfocar la imagen en el lugar que queramos para así poder eliminar los puntos de baja energía provocados por el ruido. Concretamente, si elegimos una escala muy grande el *snake* estará atraído por el borde, obteniendo un resultado muy próximo al óptimo, sin embargo si utilizamos una escala muy pequeña el *snake* estará atraído por microestructuras o elementos como el ruido cercano al contorno óptimo, pero sin llegar a alcanzar los bordes reales. Es por esto por lo que se empieza con escalas muy grandes para obtener un resultado aproximado y después se va disminuyendo dicha escala para obtener una solución mejor y más eficiente. [16]

Para definir la energía de terminación debemos encontrar las terminaciones de líneas y de las esquinas basándose principalmente en la suavidad de la imagen que vemos a continuación:

$$C(x(s), y(s)) = G_{\sigma}(x(s).y(s)) * I(x(s).y(s)),$$

a partir de esto y con el grado de curvatura de nivel de la imagen suavizada, es decir el grado de inclinación, que lo representamos como:

$$\alpha = \tan^{-1}\left(\frac{\frac{\partial}{\partial y} C(x(s).y(s))}{\frac{\partial}{\partial x} C(x(s).y(s))}\right),$$

además sabiendo que el vector unitario a la largo y perpendicular a la dirección del gradiente son respectivamente:

$$n_1(s) = (\cos \alpha(s). \sin \alpha(s))$$

$$n_2(s) = (-\sin \alpha(s). \cos \alpha(s))'$$

quedando la curvatura de nivel del contorno como la energía de terminación:

$$\begin{aligned} E_{\text{int}}(r(s)) &= \frac{\partial \alpha(s)}{\partial n_2(s)} = \frac{\frac{\partial^2 C}{\partial n_2^2}}{\frac{\partial C}{\partial n_1}} = \frac{\frac{\partial^2 C}{\partial y} \left(\frac{\partial C}{\partial x}\right)^2 - 2 \frac{\partial C}{\partial x} \frac{\partial^2 C}{\partial x \partial y} \frac{\partial C}{\partial y} + \frac{\partial^2 C}{\partial x^2} \left(\frac{\partial C}{\partial y}\right)^2}{\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}\right)^{\frac{3}{2}}} = \\ &= \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{\frac{3}{2}}} \end{aligned}$$

con la combinación de las energías de terminación y de bordes podemos crear una *snake* que se sienta atraído por ambos elementos, o alguno de ellos. El contorno del *snake* entre las líneas y los bordes en una ilusión está determinado por la suavidad de las terminaciones, y esta misma serpiente puede encontrar bordes tradicionales en imágenes naturales. [17]

Los *snakes* tienen las propiedades de comprobar experimentalmente la capacidad del *snake* para segmentar contornos subjetivos y la propiedad de histéresis. Estas dos propiedades no existían en los algoritmos clásicos de segmentación y, por lo tanto, esto es algo característico de los contornos activos.

La histéresis es la tendencia de un material a conservar alguna de sus propiedades, en ausencia del estímulo que lo ha producido.

Minimizando la energía, que es lo que realiza constantemente los *sankes*, puede reflejar la histéresis. Tiene más dificultad reflejarla en imágenes fijas, pero en la visión humana es algo muy característico correspondiente a la recreación de estímulos en movimiento.

Para finalizar veremos la energía externa, que viene dada por:

$$E_{ext} = E_{img} + E_{res},$$

donde el primer término es la energía de la imagen, que vimos anteriormente, y el segundo lo veremos a continuación.

Normalmente será introducida por el usuario y se puede definir de la siguiente forma:

$$E_{res} = E_{muse} + E_{vol},$$

la primera es lo mismo que conectar un muelle desde un punto cualquiera del *snake* hasta otro punto del *snake* o de la imagen. La podemos definir en función del parámetro de restauración del muelle y de las coordenadas de los extremos del muelle p_1, p_2 , por lo que podríamos definirlo de la siguiente forma:

$$E_{muse} = -k(p_1 - p_2),$$

la segunda dependiente de el parámetro r conocido como la distancia hasta el centro del volcán, esta energía es capaz de generar una repulsión de tipo $\frac{1}{r}$, que es truncada cercano a $r = 0$, es por esto por lo que se le conoce como nombre de volcán. Dicho volcán repelerá al *snake* de su centro sea cual sea los puntos de la imagen en el que está situado.

Aunque no lo hayamos comentando en todos los casos el *snake* estará atraído por alguna restricción en particular, pero también sigue sujeto al resto de restricciones. [7.1.7]

7.1.3. Optimización del modelo. Minimización de la energía del *snake*

Para comenzar recordamos algunas expresiones vistas anteriormente como la función de energía que viene dado por la ecuación:

$$E_{snake}^*(r) = \int_0^1 E_{int}(r(s))ds + \int_0^1 E_{img}(r(s))ds + \int_0^1 E_{res}(r(s))ds,$$

para obtener los mínimos locales de la función anterior partimos de la siguiente función:

$$E_{ext} = E_{img} + E_{res},$$

cuando $\alpha(s)$ y $\beta(s)$ son constantes obtenemos dos ecuaciones independiente de *Euler-Lagrange*:

$$\begin{aligned} -\frac{d}{ds}(\alpha(s)x''(s)) + \frac{d}{ds^2}(\beta(s)x'''(s)) + \frac{\partial E_{ext}}{\partial x} &= 0 \\ -\frac{d}{ds}(\alpha(s)y''(s)) + \frac{d}{ds^2}(\beta(s)y'''(s)) + \frac{\partial E_{ext}}{\partial y} &= 0 \end{aligned}$$

Sin embargo cuando α y β no son constantes lo más sencillo es utilizar la formulación discreta:

$$E_{snake}^*(r) = \sum_{k=1}^K E_{int}(k) ds + \sum_{k=1}^K E_{ext}(k) ds$$

Donde la energía interna que anteriormente teníamos:

$$E_{int}(r(s)) = \frac{1}{2}(\alpha |r'(s)|^2 + \beta |r''(s)|^2) = \frac{1}{2}(\alpha \left\| \frac{d\bar{r}}{ds}(s) \right\|^2 + \beta \left\| \frac{d^2\bar{r}}{ds^2}(s) \right\|^2),$$

donde el contorno discreto es $r_i = (x_i, y_i) = (x(ih), y(ih))$, utilizando el método diferencias finitas para aproximar las derivadas, la energía interna de forma discreta quedaría de la siguiente forma:

$$E_{int} = \alpha_i \frac{|r_i - r_{i-1}|^2}{2h^2} + \beta_i \frac{|r_{i-1} - 2r_i + r_{i+1}|^2}{2h^4},$$

si volvemos a aproximar las derivadas según el método de diferencias finitas si no puede resolverse analíticamente, y habiendo definido el contorno discreto anteriormente, las correspondiente ecuaciones de *Euler-Lagrange* serían:

$$\alpha_i(r_i - r_{i-1}) - \alpha_{i+1}(r_{i+1} - r_i) + \beta_{i-1}(r_{i-2} - 2r_{i-1} + r_i) - 2\beta_i(r_{i-1} - 2r_i + r_{i+1}) + \beta_{i+1}(r_i - 2r_{i+1} + r_{i+2}) + (f_x(i), f_y(i)) = 0$$

donde $r_0 = r_k$ si: $f_x(i) = \frac{\partial E_{ext}(i)}{\partial x}$, y $f_y(i) = \frac{\partial E_{ext}(i)}{\partial y}$.

En forma matricial, donde A es una matriz pentadiagonal, que solamente tiene entradas distintas a cero en la diagonal principal, y las primeras dos diagonales encima y debajo de la principal, también conocida como matriz de banda:

$$\begin{aligned} Ax + f_x(x, y) &= 0 \\ Ay + f_y(x, y) &= 0' \\ Ax + f_x(x_{t-1}, y_{t-1}) &= -\gamma(x_t - x_{t-1}) \\ Ay + f_y(x_{t-1}, y_{t-1}) &= -\gamma(y_t - y_{t-1})' \end{aligned}$$

donde γ es el valor del peso. Estas ecuaciones de *Euler* se pueden resolver mediante la matriz inversa.

[12] [16] [17]

7.1.4. Problemas y soluciones del modelo teórico

El problema del apilotonamiento surge durante la minimización de la energía, cuando el contorno se adapta al objeto que estamos segmentando. Sucede que los puntos se suelen centrar alrededor de un segmento del contorno, esto provoca que al representar el objeto segmentado tenga baja calidad en los puntos en los que la concentración de puntos es baja, por lo que puede ocasionar errores posteriores, una de las soluciones propuestas por Williams, Donna J. y Shah²³ consiguieron que los puntos distasen equitativamente unos de otros, sin embargo esto hace que el contorno inicial esté próximo al objeto. [18]

²³ Williams, Donna J. and Shah, Mubarak. "A fast algorithm for active Contours and Curvature Estimation". Computer Vision, Graphics and Image Processing.

El problema del ruido ya que los modelos deformables son muy sensibles al ruido como comenté en la parte introductoria de este trabajo. El ruido provoca que al segmentar imágenes con mucho ruido, estas no sigan un comportamiento adecuado.

Esto se puede solucionar si modificamos el modelo teórico con el que trabajamos, de forma que detectemos el ruido y una vez detectado lo incorporemos al modelo de forma que incorporaremos una nueva energía, E_{ruido} , para así poder utilizar nuestro modelo en presencia de ruido. Podemos ver un ejemplo del mismo en [18].

Unos de los problemas es el de localización del contorno inicial, que posteriormente será la solución, en el que si el contorno no está suficientemente cerca de un borde, no se verá atraído por él, mientras que si la curva no está sometida a fuerzas externas, las fuerzas internas harán que el contorno se encoja sobre sí mismo. Es por esto por lo que hay que añadir otra fuerza al contorno, para que actúe de forma más dinámica. Se va a asumir que el usuario inicializa adecuadamente los parámetros de elasticidad, rigidez, masa y rozamiento. Por ejemplo, se va a considerar el contorno como una bola en 2D que se infla o desinfla según el signo de cierto parámetro [16]. Existen muchas soluciones propuestas a este problema por diferentes autores, los cuales vamos a intentar mencionar: por Sadami²⁴, que proporciona reglas para estimar y ajustar los parámetros de amortiguamiento mediante un *snake* adaptativo que evita inestabilidades y mejora la precisión de la estimación de la forma, por Larsen y Radeva²⁵, que desarrollan un formalismo para obtener una estima de los límites superior e inferior de los condiciones de rigidez y elasticidad para obtener una segmentación óptima, por Metaxas y Kakadiaris²⁶, que propone un método eficiente e intuitivo para adaptar automáticamente los parámetros en imágenes reales donde parte del contorno puede no estar presente, por Hou y Han²⁷, que presentan un modelo mejorado que se basa en analizar la distribución del campo de fuerzas externas y encontrar dos tipos de puntos en la curva del contorno, este último consigue que el contorno no aprecie la inicialización, teniendo un amplio margen de captura, evitando que el contorno sea atrapado por un mínimo local, además de poder ajustar las concavidades, este método tiene un coste computacional menor que los métodos de flujo vectorial de gradiente [19].

Los contornos activos tienen dificultades para ajustar las concavidades de los bordes, es por ello por lo que se basan en el flujo vectorial del gradiente (GVF, *Gradient Vector Flow*) para que los snakes puedan converger en formas cóncavas. Existe un modelo posterior al GVF conocido como el flujo vectorial del gradiente generalizado (GGVF, *Generalized Gradient Vector Flow*). [16] Algunos autores que proponen soluciones a

²⁴Sadamani, R.: Adaptive snakes: control of damping and material parameters. Proc. SPIE Geometric Methods in Computer Vision, 1570, 1991.

²⁵Larsen, O. V. y P. Radeva: Calculating the Bounds on the Optimal Parameters of Elasticity for a Snake. Technical report, Aalborg University, Laboratory of Image Analysis, Denmark, 1994.

Larsen, O. V., P. Radeva y E. Martí: Guidelines for choosing optimal parameters of elasticity for snakes. Proc. Int. Conf. on Computer Analysis and Image Processing, páginas 106–113, 1995.

Larsen, O.V., P. Radeva y E. Martí: Bounds on the Optimal Elasticity Parameters for a Snake. Proc. Int. Conf. Image Analysis and Processing, páginas 37–44, 1995.

²⁶Metaxas, D. y I. A. Kakadiaris: Elastically adaptive deformable models. IEEE Trans. Pattern Anal. Machine Intelligence, 24(10):1310–1321, 2002.

²⁷Z. y C. Han: Force field analysis snake: an improved parametric active contour model. Pattern Recognition Letters, 26(5):513–526, 2005.

este problema son: Teytaud y Sarrut²⁸, se basan en resultados estadísticos de procesos empíricos para analizar el comportamiento asintótico de las *snakes* cuando se aumenta la precisión, y Blake e Isard²⁹, que desarrollaron una nueva técnica basada en las ideas de la teoría de control adaptativo y la estimación de máxima probabilidad.

Los modelos de contornos activos paramétricos no son convexos, el resultado puede ser un mínimo local cuando la energía de la curva es mínima. Para que la función que minimiza la energía genere una solución óptima, algunos autores proponen alguna variación respecto la formulación general de los contornos activos como Davatzikos y Prince³⁰.

Podemos apreciar 2 tipos de problemas más:

La discretización temporal, ya que si $f(r_{i-1})$ es muy grande, el contorno dado puede desplazarse demasiado lejos del mínimo local cercano y ya no volver. Por lo que la curva puede pasar por regiones de bordes de interés y no llegar a alcanzar el equilibrio, o estabilizarse en algún mínimo más alejado. En este problema la solución es modificarla fuerza externa normalizándola:

$$f(r_i) = \beta \frac{\nabla E_{ext}(r_i)}{\|\nabla E_{ext}(r_i)\|},$$

así cuando la curva se encuentra cercana a un borde, se encontrará atraída por éste, donde se estabilizará si no existe conflictos con la energía interna. Se podría pensar que la mejor opción es modificar el paso temporal manualmente, pero esto sólo funciona para los gradientes muy grandes, es por eso por lo que nos decantamos por normalizar la fuerza externa.

La discretización espacial, en la que conociendo valores de f en la zona discreta proporcionada por la imagen, no podemos anular f aunque tengamos cruces con f . Es decir que la solución se va a encontrar próxima a los píxeles cercanos al mínimo. La solución de este problema reside en realizar una interpolación bilineal de la energía externa E_{ext} de forma que los puntos de equilibrio serán los mínimos de E_{ext} , siendo igual para f en ciertos casos en los que su puntos serán ceros. [13]

²⁸Teytaud, O. y D. Sarrut: Convergence speed of deformable models. Proc. of Int. Joint Conf. On Neural Network, 4:2850–2855, 2001.

²⁹ Blake, A. y M. Isard: 3D position, attitude and shape input using video tracking of hands and lips. Proc. Siggraph, páginas 185–192, 1994.

³⁰Davatzikos, C. y J. L. Prince: Convexity analysis of active contour problems. Proc. Computer Vision and Pattern Recognition, páginas 674–679, 1996.

Davatzikos, C. y J. L. Prince: Adaptive active contour algorithms for extracting and mapping thick curves. Proc. Computer Vision and Pattern Recognition, páginas 524–529, 1993.

7.1.5. Comportamiento de los *snakes* tradicionales

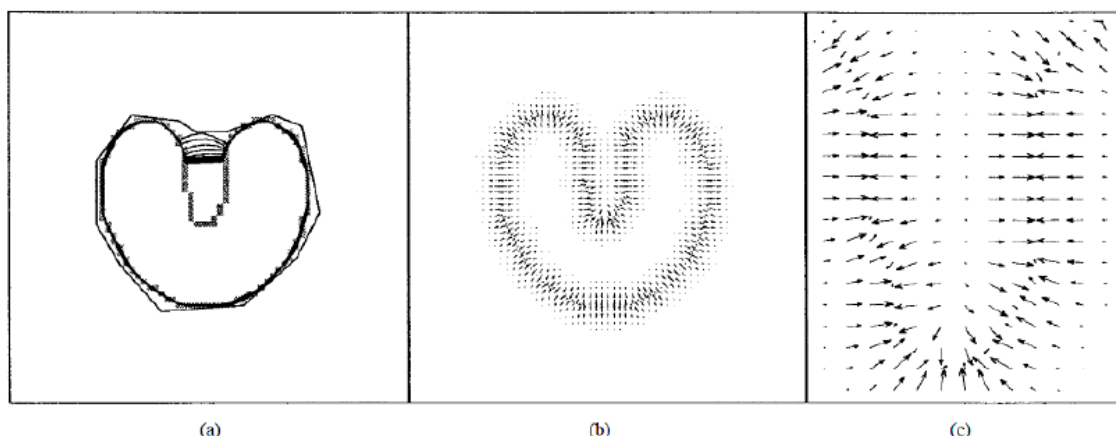


Fig. 7.1 (a) Convergencia de un *snake* usando (b) fuerzas potenciales tradicionales, y (c) muestra el centro de la concavidad l3mite.

El comportamiento de un *snake* tradicional lo podemos ver en las siguientes im3genes. En la primera de ellas, Fig. 1 (a) muestra 64x64 p3xeles de l3nea de dibujo de un objeto de U (en gris) que tiene una concavidad l3mite superior. Tambi3n se muestra una secuencia de curvas (en negro) que representa la progresi3n iterativa del *snake* tradicional inicializada fuera del objeto. En Fig. 1 (b) vemos la soluci3n final de la Fig. 1 (a), resuelta seg3n las ecuaciones de Euler utilizadas en la formulaci3n del *snake*. En esta figura podemos apreciar la magnitud de las fuerzas externas desaparecen lejos de la frontera del objeto. Variando el valor σ de podemos ampliar este rango, pero la localizaci3n del l3mite ser3 menos precisa. La Fig. 1 (c) muestra la poca convergencia de dicho *snake*, d3nde se muestran los campos de fuerza externa sin los l3mites de concavidad. Vemos que al igual que las fuerzas potenciales tradicionales en el punto dado las fuerzas son horizontalmente opuestas.

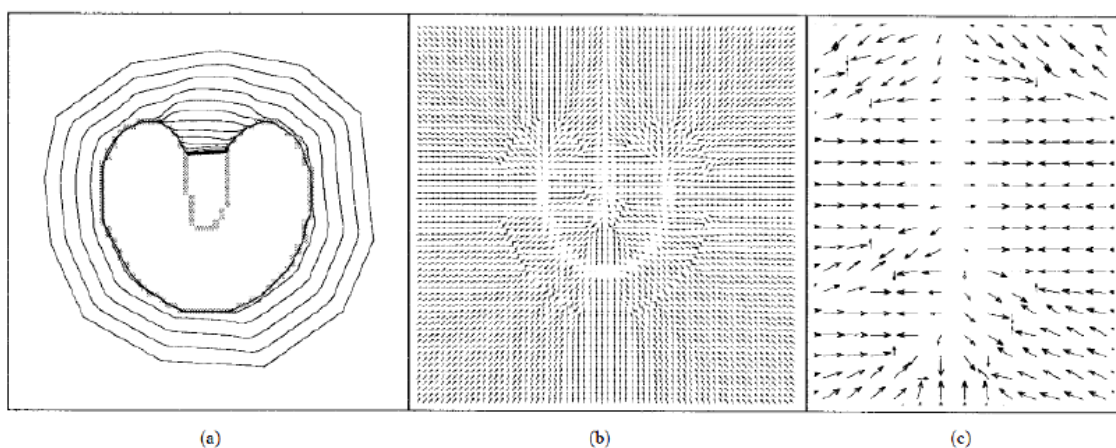


Fig. 7.2 (a) Convergencia de un *snake* usando (b) fuerzas potenciales de distancia, y (c) muestra el centro de la concavidad l3mite.

Cohen y Cohen ³¹ proponen un modelo de fuerza externa que aumenta significativamente el rango de captura tradicional del *snake*. Estas fuerzas externas son

³¹L. D. Cohen and I. Cohen "Finite-element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1131–1147, Nov. 1993

el gradiente negativo de una funci3n potencial que se calcula utilizando el mapa de distancias de Euclides. Estas fuerzas se refieren a la posible distancia entre algunas fuerzas. La Fig. 2 (a) muestra el objeto en forma de U (en gris) y unas secuencias de curvas de nivel (en negro) que representa la progresi3n de la inicializaci3n del *snake*, lejos del objeto. La distancia de las fuerzas potenciales de la Fig. 2 (b) tienen vectores con grandes magnitudes lejos del objeto, lo cual explica el rango mayor de captura para este modelo de fuerzas externas. En la Fig. 2 (a) se aprecia que este *snake* tampoco converge a la concavidad del l3mite, por lo que este problema no se soluciona con la distancia de fuerzas potenciales. En la Fig. 2 (c), vemos que al igual que las fuerzas potenciales tradicionales en el punto dado las fuerzas son horizontalmente opuestas.

7.2. Gradient Vector Flow

Nuestra *snake*, que llamamos vector gradiente de flujo (*GVF*), comienza con el c3lculo de un campo de fuerzas, fuerzas *GVF*, sobre una imagen. Las fuerzas *GVF* se utilizan para conducir el *snake* hacia los bordes de la imagen. Esta *snake* esta modelada como un objeto f3sico que tiene cierta resistencia tanto de plegado como de estiramiento. Estas fuerzas se calculan mediante la aplicaci3n de las ecuaciones generalizadas a ambos componentes del gradiente de un mapa de bordes de la imagen. Las fuerzas externas *GVF* es la caracter3stica que hace que esta *snake* sea diferente a la *snake* tradicionales. Debido a que las fuerzas *GVF* se derivan de una operaci3n de difusi3n, tienden a extenderse muy lejos del objeto. Esto permite ampliar la “zona de captura” de objetos, de esta forma las *snakes* pueden encontrar objetos que est3n muy lejos de la posici3n inicial de la *snake*. Esta difusi3n es la misma que crea fuerzas para aproximar el contorno al borde real en regiones c3ncavas. Una diferencia fundamental entre esta formulaci3n y la tradicional es que las fuerzas *GVF* no son puramente fuerzas irrotacionales. De hecho, por lo general comprenden tanto fuerzas irrotacionales como solenoidales. Esta propiedad hace que estas fuerzas no puedan ser derivadas de la minimizaci3n de energ3a de los *snakes* tradicionales.

Modelo te3rico

Partiendo de la siguiente funci3n:

$$E_{ext} = E_{img} + E_{rec},$$

cuando $\alpha(s)$ y $\beta(s)$ son constantes obtenemos dos ecuaciones independiente de *Euler-Lagrange*:

$$\begin{aligned} -\frac{d}{ds}(\alpha(s)x''(s)) + \frac{d}{ds^2}(\beta(s)x'''(s)) + -\frac{\partial E_{ext}}{\partial x} &= 0 \\ -\frac{d}{ds}(\alpha(s)y''(s)) + \frac{d}{ds^2}(\beta(s)y'''(s)) + -\frac{\partial E_{ext}}{\partial y} &= 0 \end{aligned},$$

que se puede representar de la siguiente forma:

$$\alpha x''(s) - \beta(s)x'''(s) - \nabla E_{ext} = 0,$$

para encontrar una soluci3n a la ecuaci3n anterior, el *snake* se hace din3mico mediante el tratamiento de x como funci3n del tiempo, $x(s, t)$. Para obtener la soluci3n de la ecuaci3n anterior se debe estabilizar $x(s, t)$, y es cuando $x_t(s, t)$ desaparece, y obtenemos la soluci3n de 3sta, a partir de la derivada parcial de x con respecto a t :

$$x_t(s, t) = \alpha x''(s, t) - \beta x'''(s, t) - \nabla E_{ext},$$

El flujo del vector gradiente (*GVF field*), es definido como un nuevo campo de fuerza externa rotacional. Utilizando una condición de equilibrio de fuerzas como punto de partida, permitimos que al *GVF field* pueda remplazar el campo de fuerza potencial en la ecuación anterior. Definiendo una nueva *snake*, conocida como *GVF snake*. Los puntos del campo *GVF* apuntan hacia la frontera del objeto cuando están muy cerca del límite, pero varía suavemente sobre regiones de imágenes homogéneas, hasta los bordes de la imagen. Las principales ventajas del campo *GVF* son que éstas pueden capturar una *snake* de un gran rango, de ambos lados de los límites del objeto, y que puede forzar a regiones cóncavas.

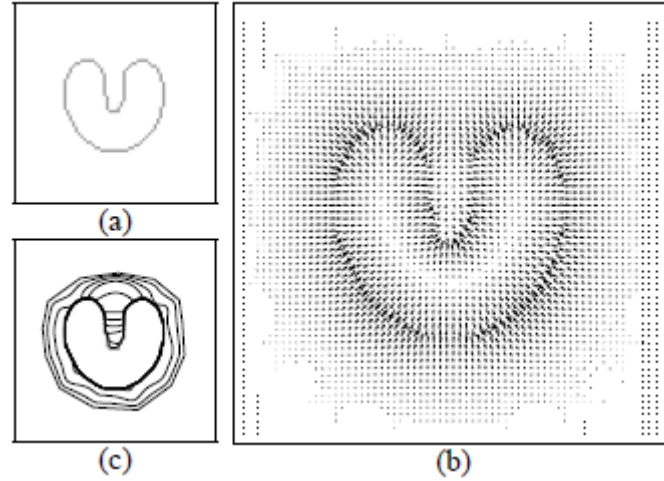


Figura 7.3: *Snake* con fuerzas *GVF* moviéndose hacia una región cóncava.

Partiendo por definir un mapa de bordes, *edge map*, $f(x, y)$ derivada de la imagen de $I(x, y)$, que será mayor cerca de los bordes de la imagen. Entonces, podemos utilizar:

$$f(x, y) = -E_{ext}^i(x, y),$$

dónde $i = 1, 2, 3, 4$. El campo ∇f tiene vectores hacia los bordes, pero no tiene un rango de captura estrecho. Por otro lado, en regiones homogéneas, $I(x, y)$ es constante, ∇f es cero, por lo que no hay información sobre bordes cercanos o lejanos disponibles. Definimos flujo del vector gradiente, *gradiente vector flow field*, como un vector campo $v(x, y) = (u(x, y), v(x, y))$, que minimiza la energía funcional:

$$\mathcal{E} = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |v - \nabla f|^2 dx dy$$

La formulación de la variación sigue un principio estándar, produciendo resultados suaves cuando no hay datos. Vemos que cuando $|\nabla f|$ es pequeño, la energía está dominada por las derivadas parciales del vector campo. Por otro lado, cuando $|\nabla f|$ es grande el segundo término domina la integral, y se minimice estableciendo, $v = \nabla f \mu$, es el parámetro que equilibra el primer y el segundo término, debe ser acorde con la cantidad de ruido que haya en la imagen.

Usando el cálculo de la variación, podemos mostrar que el *GVF* puede ser resuelto por las ecuaciones de Euler:

$$\mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0$$

$$\mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0$$

Donde ∇^2 es el operador *laplaciano*. Esta igualdad, nos da otra perspectiva acerca de la formulación de *GVF*. Cuando son regiones homogéneas, el segundo término de ambas ecuaciones son cero, a causa de que el gradiente de $f(x, y)$ es cero. En estas regiones, u y v , son determinadas por las ecuaciones de *Laplace*, que da lugar a un tipo de relleno de la información tomada en los límites de la región.

Las funciones anteriores pueden resolverse tratando u y v , como funciones de tiempo:

$$u_t(x, y, t) = \mu \nabla^2 u(x, y, t) - [(u(x, y, t) - f_x(x, y))(f_x(x, y)^2 + f_y(x, y)^2)]$$

$$v_t(x, y, t) = \mu \nabla^2 v(x, y, t) - [(v(x, y, t) - f_y(x, y))(f_x(x, y)^2 + f_y(x, y)^2)]$$

Teniendo en cuenta que las ecuaciones están desacopladas, pueden ser resueltas como ecuaciones diferenciales parciales separadas de forma escalar en u y v . Las últimas ecuaciones son conocidas como las ecuaciones generalizadas de difusión. Después de computar $v(x, y)$, reemplazamos la fuerza $-\nabla E_{ext}$, de la ecuación dinámica del *snake*:

$$x_t(s, t) = \alpha x''(s, t) - \beta(s)x'''(s, t) - \nabla E_{ext}$$

quedando la ecuación conocida como *GVF snake*:

$$x_t(s, t) = \alpha x''(s, t) - \beta(s)x'''(s, t) + v$$

Tener en cuenta, que se puede generalizar los *GVF* a tres dimensiones, e implementarlo sobre superficies deformables. [19][20][21]

323334353637383940414243

³²<http://www.iacel.ece.jhu.edu/static/gvf/>

³³C. Xu and J.L. Prince, "Gradient Vector Flow: A New External Force for Snakes," Proc. IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR), Los Alamitos: Comp. Soc. Press, pp. 66-71, June 1997. [[pdf \(2.5MB\)](#)].

³⁴C. Xu and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE Transactions on Image Processing*, 7(3), pp. 359-369, March 1998. [[pdf \(0.4MB\)](#)].

³⁵C. Xu and J. L. Prince, "Generalized Gradient Vector Flow External Forces for Active Contours," *Signal Processing --- An International Journal*, 71(2), pp. 131-139, December 1998. [[pdf \(0.5MB\)](#)].

³⁶C. Xu and J. L. Prince, "Global Optimality of Gradient Vector Flow", *Proc. of 34th Annual Conference on Information Sciences and Systems (CISS'00)*, Princeton University, March 2000. [[pdf \(0.7MB\)](#)].

³⁷C. Xu, D. L. Pham, and J. L. Prince, "Medical Image Segmentation Using Deformable Models," *SPIE Handbook on Medical Imaging -- Volume III: Medical Image Analysis*, edited by J.M. Fitzpatrick and M. Sonka, May 2000. [[pdf \(5.6MB\)](#)].

³⁸C. Xu and J. L. Prince, "Gradient Vector Flow Deformable Models", *Handbook of Medical Imaging*, edited by Isaac Bankman, Academic Press, September, 2000. [[pdf \(1.7MB\)](#)].

³⁹C. Xu, D. L. Pham, M. E. Rettmann, D. N. Yu, and J. L. Prince, "Reconstruction of the Human Cerebral Cortex from Magnetic Resonance Images," *IEEE Transactions on Medical Imaging*, 18(6), pp. 467-480, June, 1999. [[pdf \(0.9MB\)](#)].

⁴⁰C. Xu, A. Yezzi, Jr., and J. L. Prince, "On the Relationship between Parametric and Geometric Active Contours", in Proc. of 34th Asilomar Conference on Signals, Systems, and Computers, pp. 483 -489, October 2000. [[pdf \(0.2MB\)](#)].

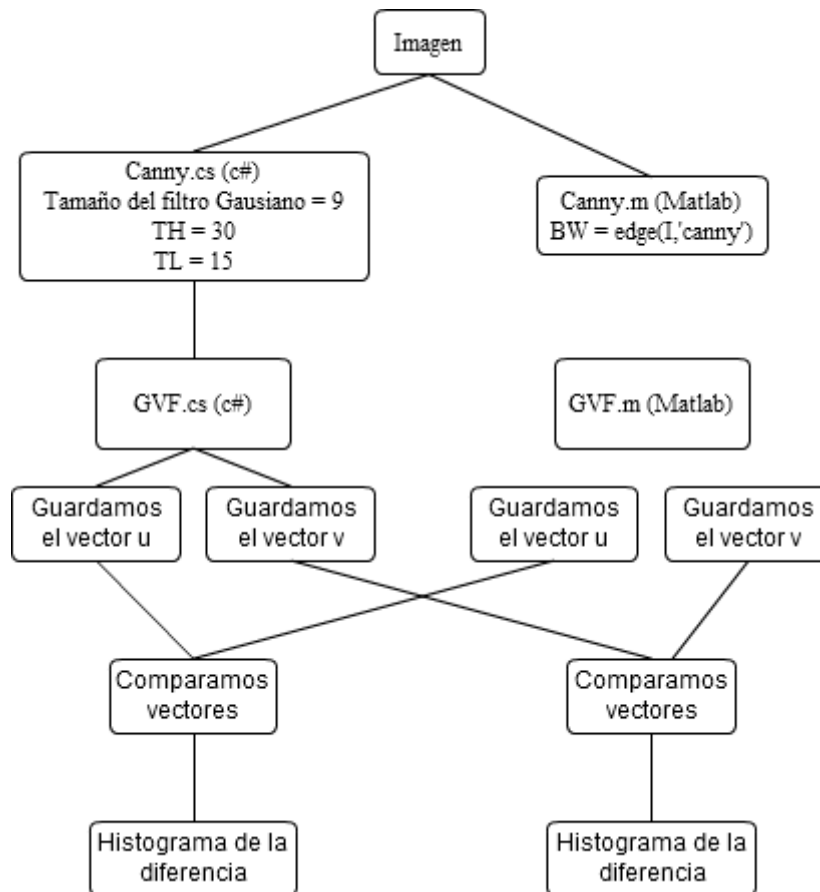
⁴¹C. Xu, A. Yezzi, Jr., and J. L. Prince, "A Summary of Geometric Level-Set Analogues for a General Class of Parametric Active Contour and Surface Models", in Proc. of 2001 IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM 2001), pp. 104-111, July 2001. [[pdf \(0.3MB\)](#)].

⁴²C. Xu, *Deformable Models with Application to Human Cerebral Cortex Reconstruction from Magnetic Resonance Images*, PhD Dissertation, Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, 21218, USA. [[pdf \(20MB\)](#)].

⁴³X. Han, C. Xu, and J. L. Prince, "A Topology Preserving Level Set Method for Geometric Deformable Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 25, No 6, pp. 755-768, June 2003. [[pdf \(1.8MB\)](#)].

7.2.1. Resultados GVF

Para poder afirmar que se est1 aplicando correctamente el algoritmo, se ha decidido comparar los vectores obtenidos tras la aplicaci3n de este m3todo en nuestra aplicaci3n, con los mismos vectores obtenidos tras ejecutar el c3digo de este mismo m3todo que el autor permite descargar en su web en otro lenguaje, matlab.⁴⁴ Vemos a continuaci3n un diagrama del procedimiento seguido para comparar los vectores en diferentes lenguajes.



Se ha decidido utilizar los mismos par1metros que se utilizan en otros lenguajes, adem1s son lo recomendados por el autor para este tipo de im1genes y para poder realizar una comparaci3n directa entre el algoritmo implementado y el original. Estos par1metros son *mu*, el par1metro de ponderaci3n, que suele variar entre 0.01 y 0.2, e *iter*, que es el n1mero de iteraciones, que por lo general oscila entre 20 y 100. Los par1metros que hemos utilizado para nuestra comparaci3n se pueden ver en la siguiente tabla.

Par1metros de GVF	
Mu	0.2
Iter	80

Tabla 7.1: Par1metros utilizados en el algoritmo GVF

⁴⁴<http://www.iac1.ece.jhu.edu/static/gvf/>

A continuaci3n podemos ver algunas de las im3genes obtenidas, adem3s de los histogramas obtenidos tras la comparaci3n de los vectores antes mencionados.

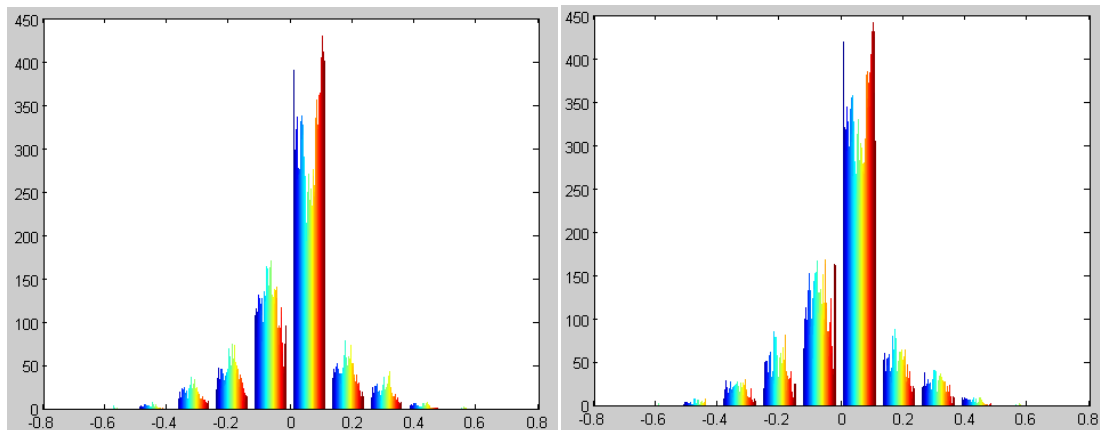


Figura 7.4: 287269_241852_D.bmp, imagen a la izquierda, histograma resultante de la diferencia en el vector u entre su implementaci3n Matlab y la realizada en este proyecto. Imagen a la derecha histograma resultante v.

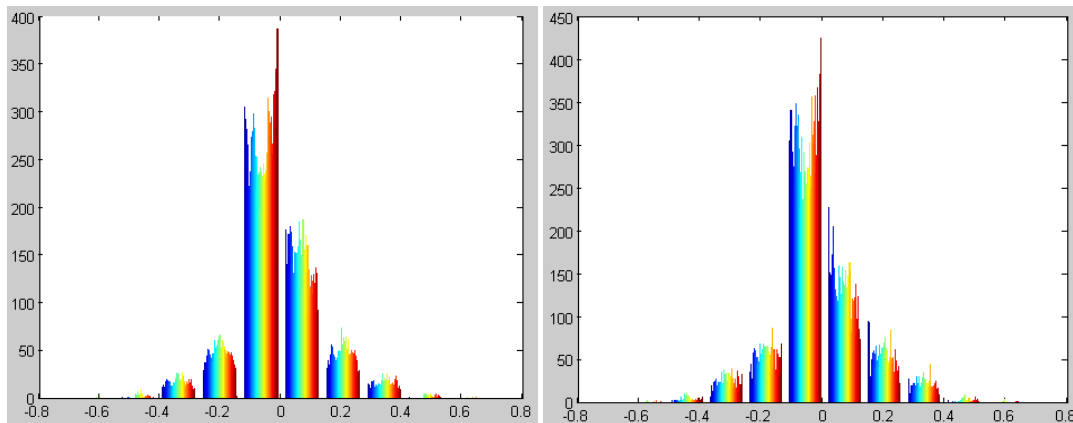


Figura 7.5: 288615_240875_I.bmp, imagen a la izquierda, histograma resultante tras comparar vector u. Imagen a la derecha histograma resultante v.

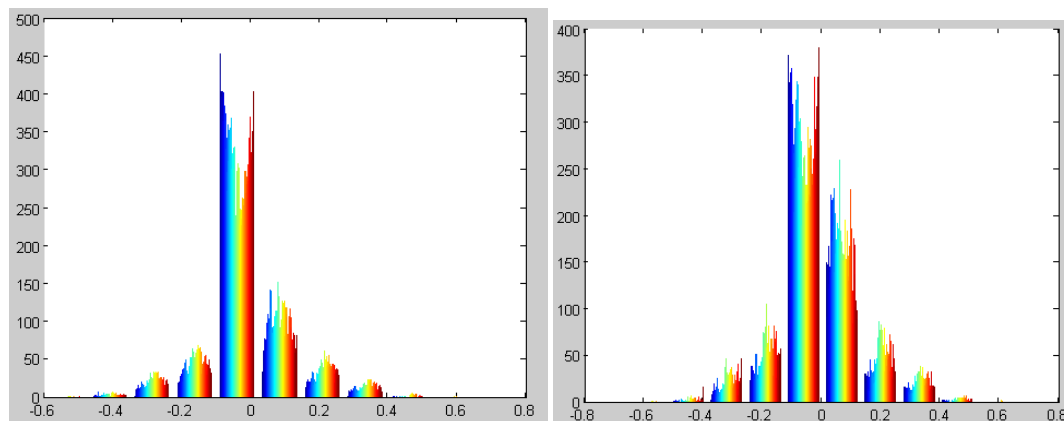


Figura 7.6: 288615_239804_I.bmp, imagen a la izquierda, histograma resultante tras comparar vector u. Imagen a la derecha histograma resultante v.

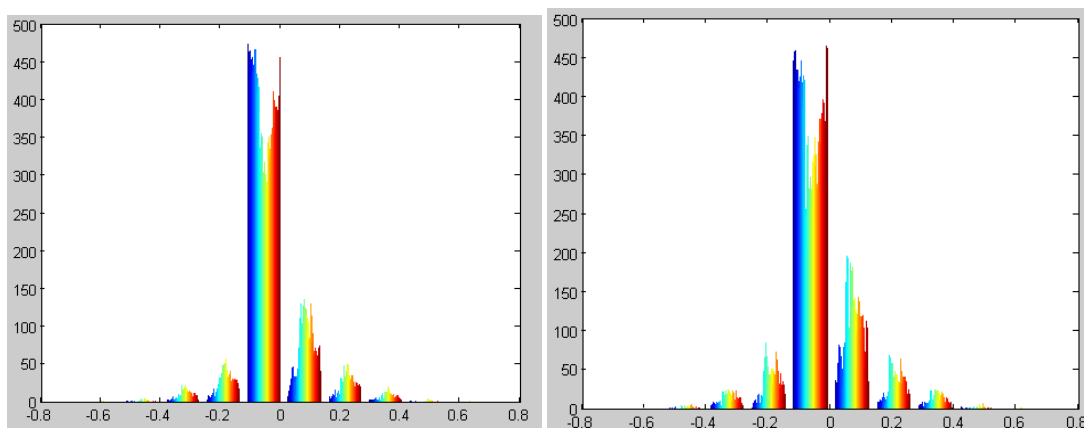


Figura 7.7: 288615_239279_I.bmp, imagen a la izquierda, histograma resultante tras comparar vector u. Imagen a la derecha histograma resultante v.

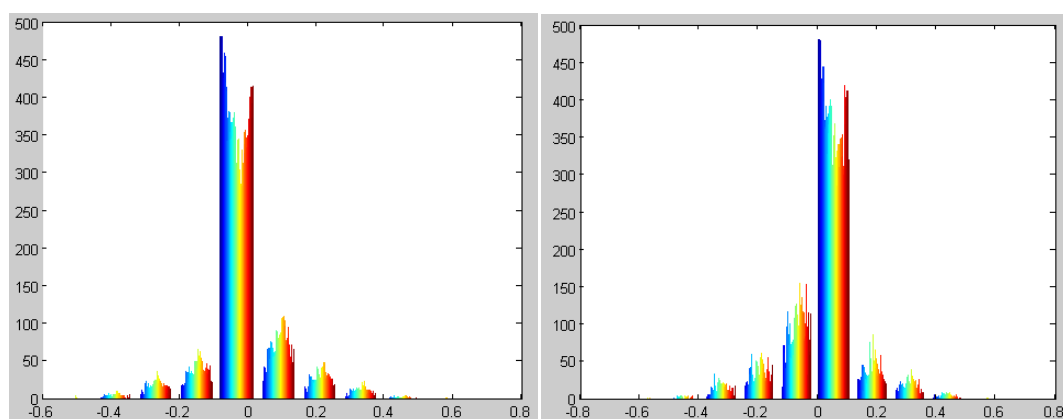


Figura 7.8: 287810_245444_I.bmp, imagen a la izquierda, histograma resultante tras comparar vector u. Imagen a la derecha histograma resultante v.

7.2.2. Conclusi3n

Los resultados obtenidos los podemos considerar satisfactorios. Los resultados de la comparaci3n de los vectores u, v, ejecutados en diferentes lenguajes nos deja una variaci3n que en la mayor3a de los casos ronda entre 0, y 0,1. Este error, al m3nimo, no resulta significativa entre la implementaci3n en Matlab y la que se ha realizado en este trabajo. Adem3s, hay que tenerlo muy en cuenta que puede ser debido al error arrastrado por m3todos anteriores. En nuestro caso, hemos aplicado antes el m3todo de Canny, el cual tiene par3metros ajustables que hace que aplicar diferentes valores origine resultados diferentes. Por ejemplo, en las librer3as de Matlab, existe la posibilidad de realizar un ajuste autom3tico del umbral, tambi3n permite ajuste manual, pero el rango de valores es diferente, por lo que es complicado realizar un entorno de pruebas adecuado. Un m3todo parecido a este ajuste de umbral autom3tico hemos intentado aplicar en nuestro estudio sin resultados satisfactorios (ver cap3tulo 5, apartado 2).

Adem3s, el mismo m3todo acepta distintos par3metros. En la propia web del autor vemos como en cada lenguaje se ajusta a las posibilidades de este. En Matlab, conseguimos una manejo con las matrices muy sencillo, sin embargo no se consiguen buenos resultados respecto al tiempo de ejecuci3n. En c# el manejo de matrices, no es tan sencillo, se pueden hacer uso de diferentes librer3as para facilitarlas, pero en nuestro caso, se ha decidido, partir de la implementaci3n en c del autor, que facilita el uso de matrices, con diferentes operaciones con vectores multidimensionales.

En tiempos de ejecución se han conseguido resultados extraordinarios como era de esperar. Sin embargo el error que cometemos se puede minimiza, o bien realizando una modificación de nuestro código con dicho fin, o bien intentando hacer uso de librerías que implementen las operaciones necesarias. Aunque pueda ser conveniente implementarlo, hacerlo incrementaría mucho el coste de este trabajo, por lo que quedaría pendiente para trabajos futuros.

7.2.3. Resultados *snake*

Tras ver que los resultados obtenidos por el algoritmo GVF se parecían mucho a los obtenidos por otras implementaciones en otros lenguajes, se ha decidido implementar la última parte de este proyecto, los *snakes*. Podemos ver la descripción teórica a lo largo del capítulo 6.

Como hemos podido ver en algún diagrama anterior (Ver diagrama 3), antes de aplicar los contornos activos, realizamos un fase de preprocesado, en la que en la que se detecta el radio del iris, y de la pupila, además del centro de estos. Después de esto ejecutamos dos algoritmos, para intentar resaltar los bordes deseados, con el fin que la detección de los bordes del iris sea más fácil. En la implementación de los *snake* hemos partido de la base de que los contornos iniciales se obtienen de la fase de preprocesado.

A continuación vemos el diagrama de flujos que hemos comentado anteriormente:

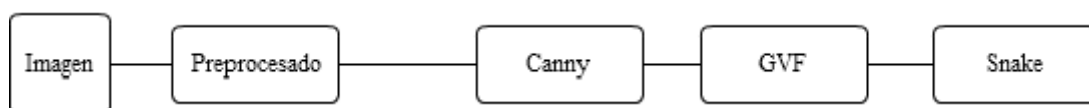


Figura 7.9: Diagrama

En nuestra implementación nos hemos basado en una librería para implementar este algoritmo. Emgu CV es una plataforma cruzada que se creó a partir de OpenCV para el procesamiento de imágenes. Permite que las funciones de OpenCV puedan ser llamadas desde .NET. Los lenguajes compatibles son algunos como: C#, VB, VC ++. Todas las pruebas se han realizado con la versión opensource, pero dispone de versión con licencia comercial.⁴⁵

Para empezar se ha decidido partir de algunas imágenes sencillas para ver que se ajusta correctamente.

En estas imágenes iniciales, con formas geométricas, vemos como tenemos que variar los parámetros de los filtros de Canny y GVF, además de jugar con aplicarlos o no y ajustar los parámetros del *snake*, para conseguir que el contorno se ajuste satisfactoriamente a las imágenes dadas.

⁴⁵EmguCV - http://www.emgu.com/wiki/index.php/Main_Page

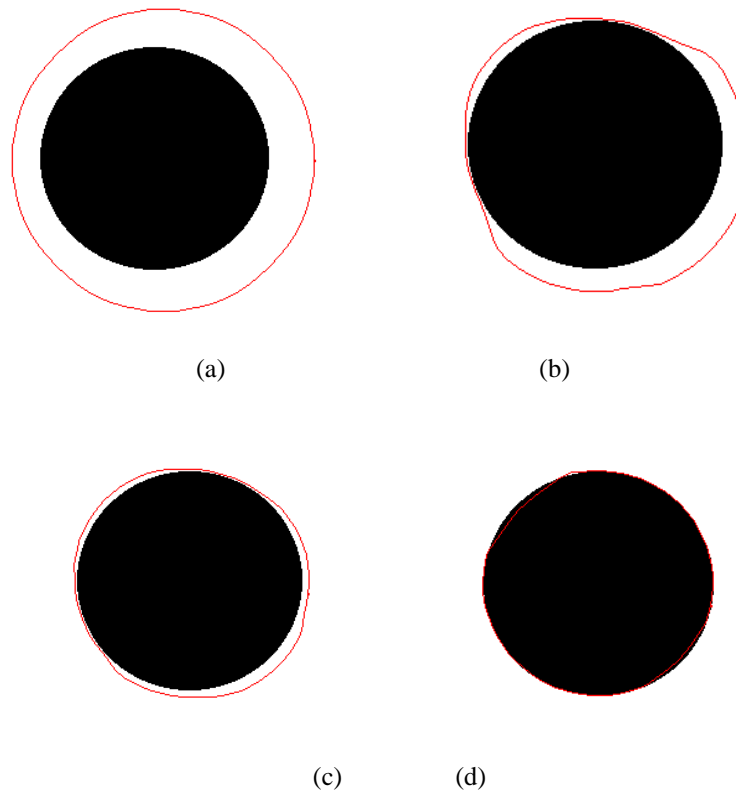


Figura 7.10: Imagen circular de prueba, (a) imagen original con el contorno inicial. (b) imagen tras ejecutar el snake, (Umbrales Canny: 30, 10. GVF: iter = 80; $\mu=0.2$. Tiempo de ejecuci3n: 2.973 segundos). (c) imagen tras ejecutar el snake, (Umbrales Canny: 30, 10. GVF: iter=10; $\mu=0.1$. Tiempo de ejecuci3n: 2.897 segundos). (d) imagen tras ejecutar el snake sin aplicar GVF (Umbrales Canny: 30, 10. Tiempo de ejecuci3n: 2.7613 segundos).

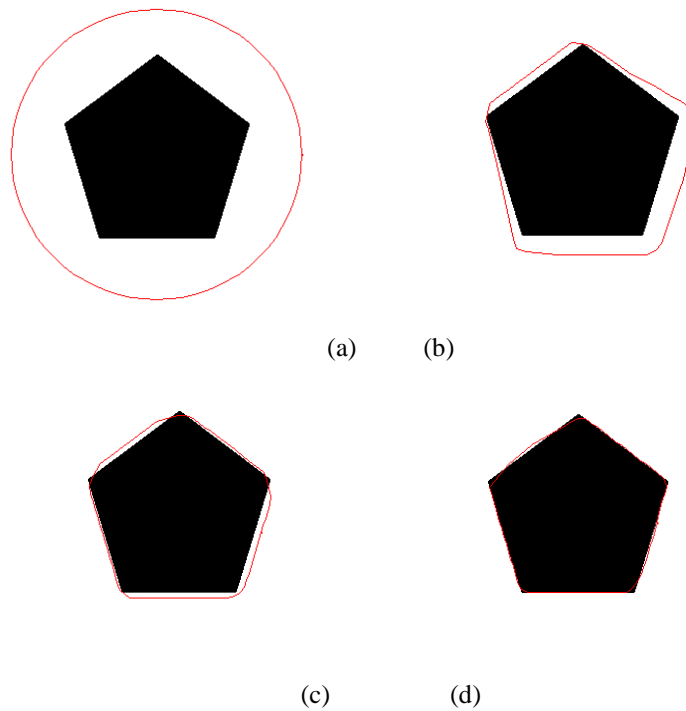


Figura 7.11: Imagen pentagonal de prueba, (a) imagen original con el contorno inicial. (b) imagen tras ejecutar el snake, (Umbrales Canny: 30, 10. GVF: iter = 80; $\mu=0.2$. Tiempo de ejecuci3n:

2.7863 segundos). (c) imagen tras ejecutar el snake, (Umbrales Canny: 30, 10. GVF: iter=10; mu=0.1. Tiempo de ejecuci3n: 2.932 segundos). (d) imagen tras ejecutar el snake sin aplicar GVF (Umbrales Canny: 30, 10. Tiempo de ejecuci3n: 2.905 segundos).

Para nuestro estudio ya se han fijado los par3metros adecuados de los filtros descritos en cap3tulos anteriores, por lo haremos nuestro estudio modificando s3lo los que corresponde al *snake*.

Los par3metros fijados anteriormente son:

Par3metros de Canny	
Tama3o del filtro <i>Gaussiano</i>	9
Umbral alto	30
Umbral bajo	10

Tabla 7.2: Par3metros utilizados en los resultados del algoritmo de Canny

Par3metros de GVF	
Mu	0.2
Iter	80

Tabla 7.3: Par3metros utilizados en el algoritmo GVF

En la fase de procesado, se obtiene el radio del iris, y de la pupila, junto con su centro. A partir de estos valores se generan los contornos iniciales. Vemos algunos ejemplos:

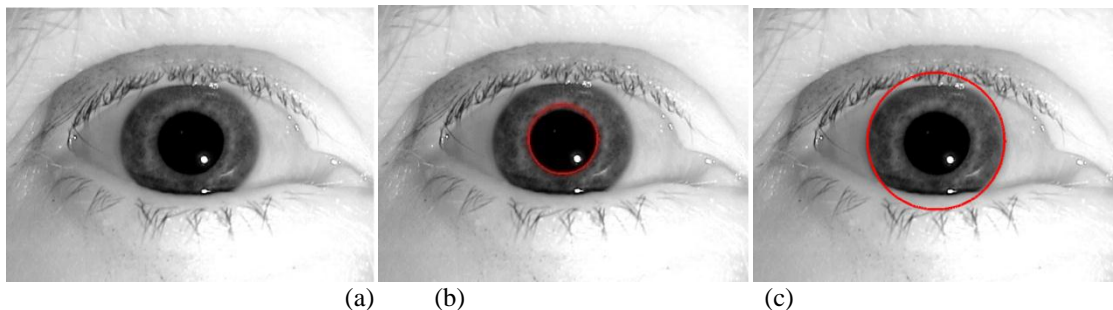


Figura 7.12: 287269_241852_D.bmp (a) imagen original. (b) imagen con los contornos iniciales para el borde interno. (c) imagen con los contornos iniciales para el borde externo.

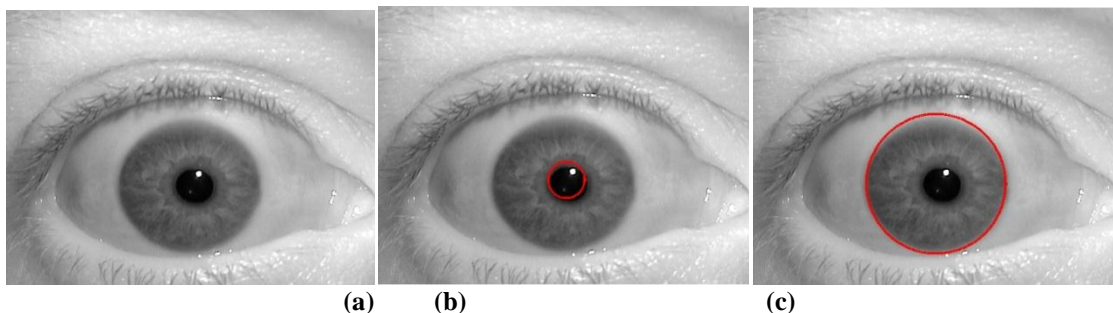


Figura 7.13: 287810_243007_D.bmp (a) imagen original. (b) imagen con los contornos iniciales para el borde interno. (c) imagen con los contornos iniciales para el borde externo.

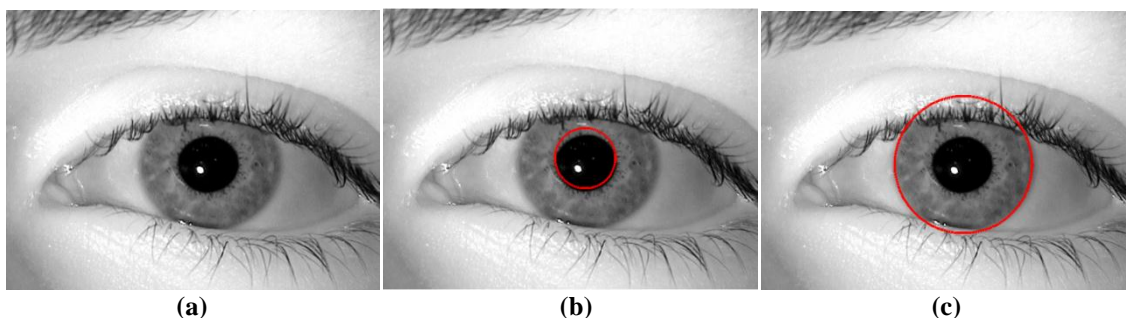


Figura 7.14: 290821_240477_I.bmp (a) imagen original. (b) imagen con los contornos iniciales para el borde interno. (c) imagen con los contornos iniciales para el borde externo.

Como hemos visto, en los ejemplos la fase de pre-procesado, nos da unos resultados muy buenos, que nos permite pintar los contornos iniciales.

Los parámetros con los que se ha ido ajustando el contorno para detectar el borde interno del iris varían entre los siguientes valores:

Alpha	0.05 - 5.55
Beta	0 - 2
Gamma	1 - 2
num. de iter	50

Tabla 7.4: Rango de valores para detectar el borde interno del iris mediante el *snake*.

De la base de datos utilizada en este proyecto se han evaluado alrededor de 500 imágenes, de las cuales a continuación se muestran algunos de los resultados obtenidos para la detección del borde interno. En estos resultados se muestran los contornos resultantes al aplicar diferentes parámetros.

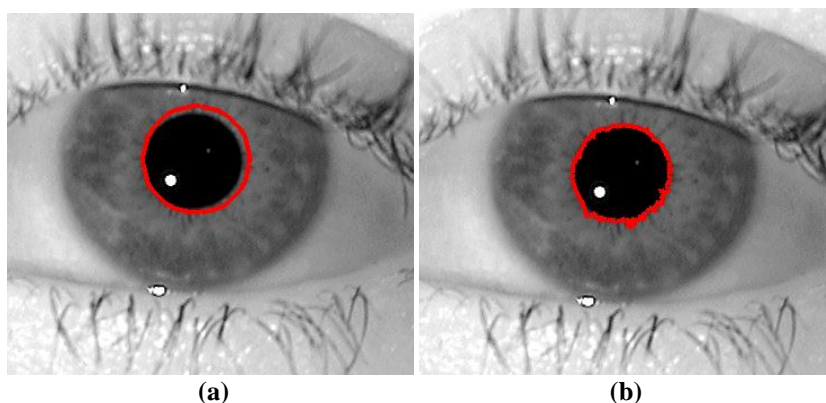


Figura 7.15: 290821_240477_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. (alpha = 0.55, beta = 0, gamma = 1, iteraciones = 50. Tiempo de ejecución = 2.807 segundos)

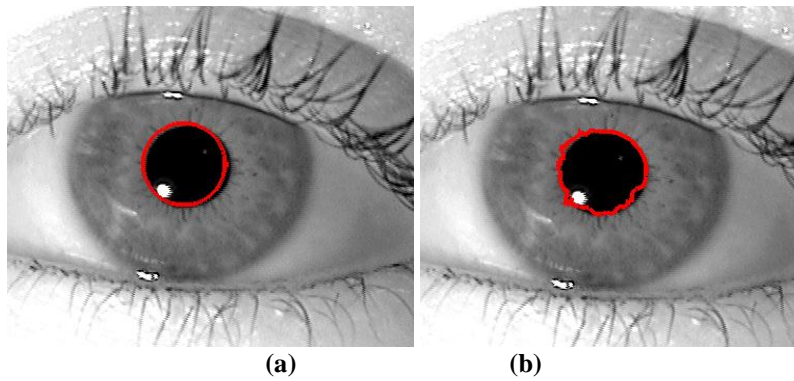


Figura 7.16: 288615_245178_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 0.55$, $\beta = 0$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecuci3n = 2,691 segundos)

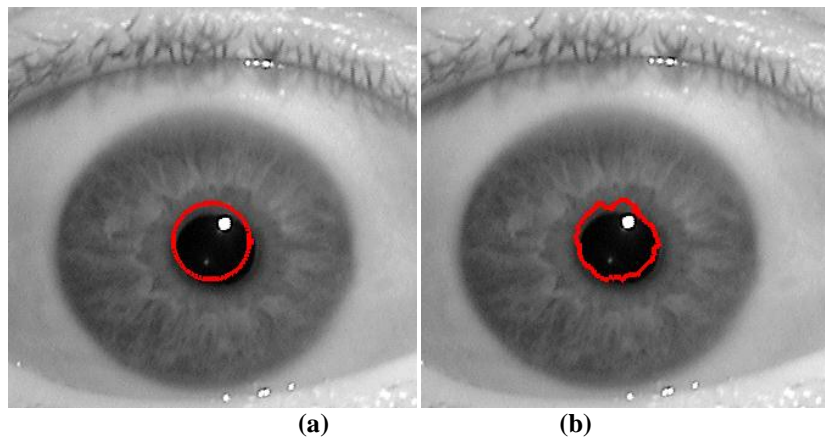


Figura 7.17: 287810_243007_D.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 5.55$, $\beta = 0$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecuci3n = 2.611 segundos)

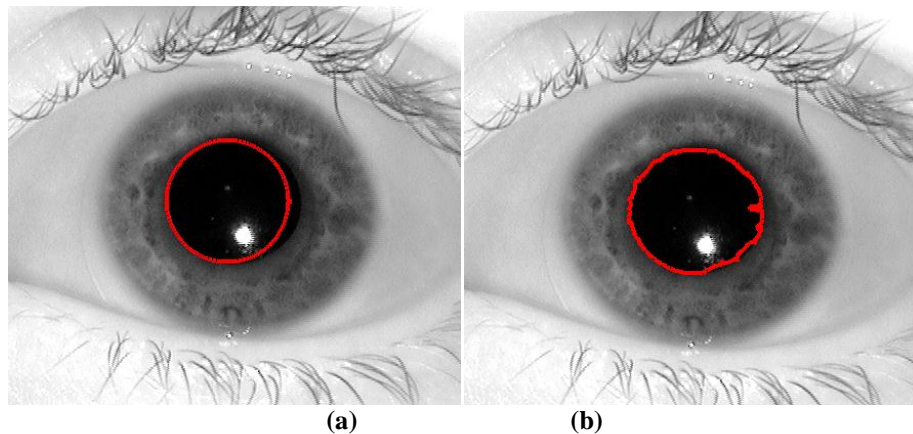


Figura 7.18: 287810_243007_D.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 0.55$, $\beta = 0$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecuci3n = 2.658 segundos)

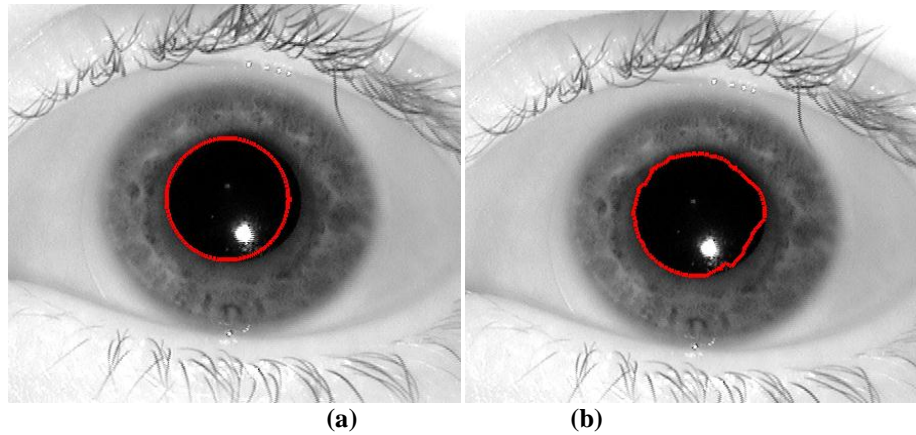


Figura 7.19: 287810_243007_D.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 0.55$, $\beta = 2$, $\gamma = 2$, iteraciones = 50. Tiempo de ejecuci3n = 3,011 segundos)

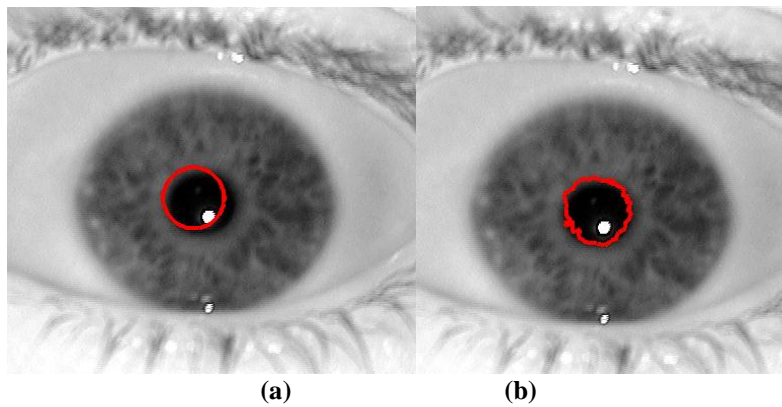


Figura 7.20: 291008_242007_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 0.05$, $\beta = 0$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecuci3n = 2.619 segundos)

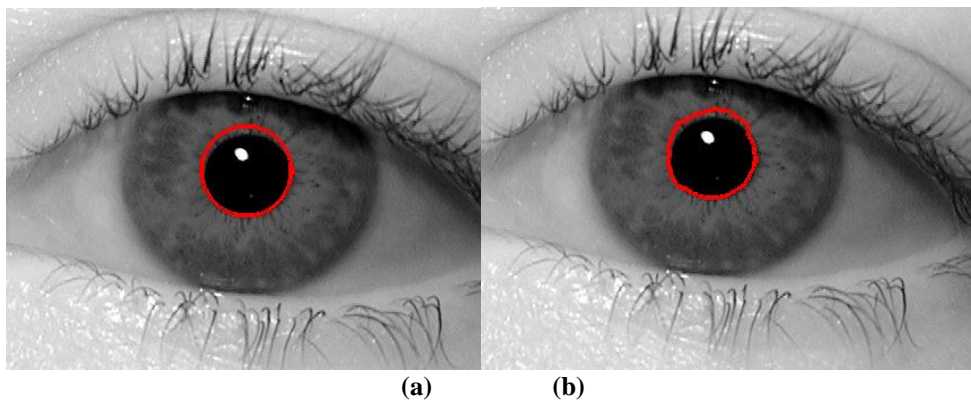


Figura 7.21: 288615_246050_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 0.55$, $\beta = 1$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecuci3n = 2,874 segundos)

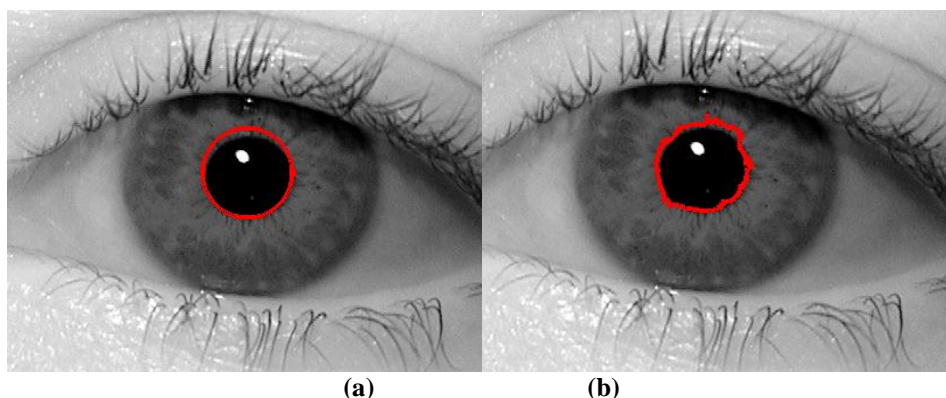


Figura 7.22: 288615_246050_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 0.55$, $\beta = 1$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecuci3n = 2,766 segundos)

A continuaci3n podemos ver un breve an3lisis de los datos obtenidos:

Tiempo de ejecuci3n	
Media	2.794 seg

Tabla 7.5: Tiempos de ejecuci3n

Debido a la gran variedad de las im3genes de entrada, se ha decidido realizar un estudio de diferentes umbrales. Para ello se han considerado un total de 500 im3genes elegidas aleatoriamente de la base de datos. A continuaci3n vemos una tabla del an3lisis realizado para cada una de las parejas de umbrales.

		<i>snake</i> - borde interno						
Alpha		0.05	0.55	5.55	0.05	0.55	0.55	5.55
Beta		0	0	0	1	2	2	1
Gamma		1	1	1	1	1	2	2
num. de iter		50	50	50	50	50	50	50
Borde	Adecuado	92%	91%	80%	75%	80%	79%	85
	Posibles errores	7%	7%	18%	23%	18%	19%	13%
	Malo	1%	2%	2%	2%	2%	2%	2%

Tabla 7.6: An3lisis de los resultados despu3s ejecutar el *snake* para el borde interno.

Los parámetros con los que se ha ido ajustando el contorno para detectar el borde externo del iris varían entre los siguientes valores:

Alpha	0.05 - 15
Beta	0 - 2
Gamma	1 - 2
num. de iter	50

Tabla 7.7: Rango de valores para detectar el borde externo del iris mediante el *snake*.

Al igual que para el borde interno se han evaluado alrededor de 500 imágenes, de las cuales a continuación se muestran algunos de los resultados obtenidos para la detección del borde interno. En estos resultados se muestran los contornos resultantes al aplicar diferentes parámetros.

Para el borde exterior, también se ha usado los resultados obtenidos en el preprocesado para pintar los contornos iniciales.

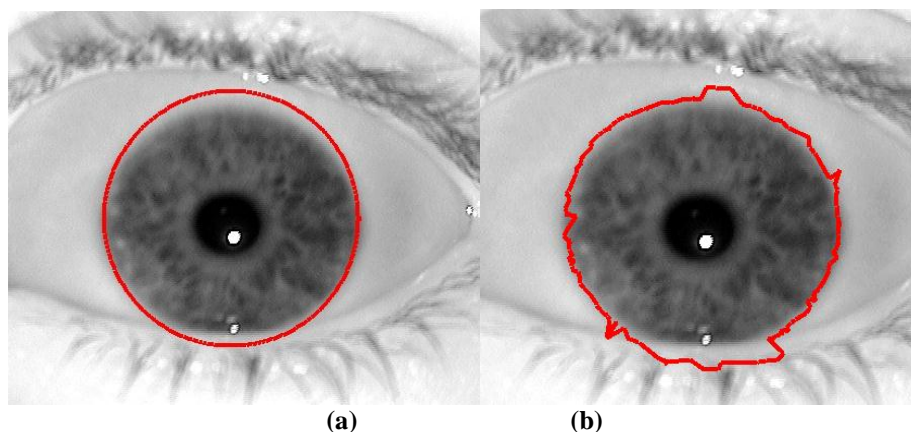


Figura 7.23: 291008_242007_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. (alpha = 0.05, beta = 0, gamma = 1, iteraciones = 50. Tiempo de ejecución = 2,768 segundos)

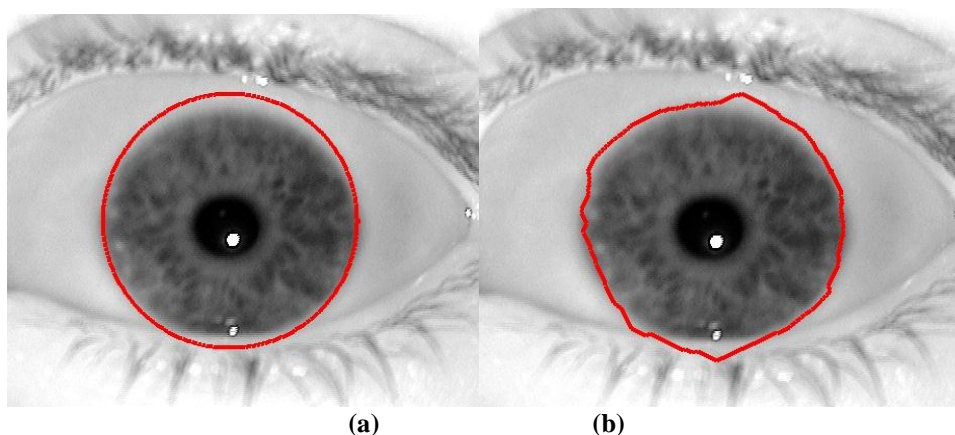


Figura 7.24: 291008_242007_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. (alpha = 5, beta = 1, gamma = 1, iteraciones = 50. Tiempo de ejecución = 2649 segundos)

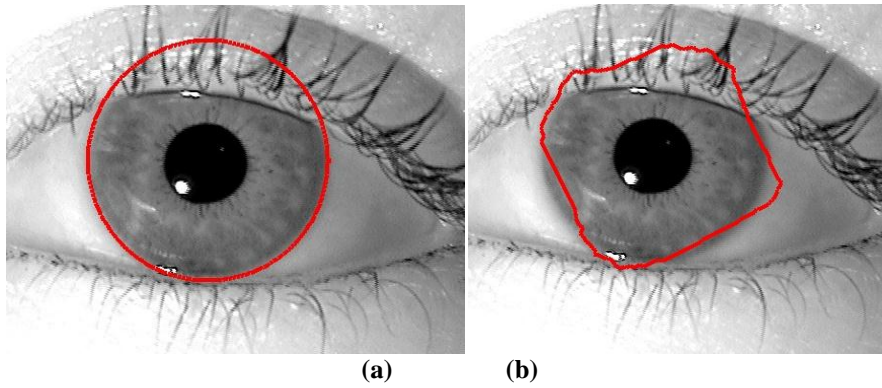


Figura 7.25: 288615_245178_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 15$, $\beta = 0$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecución = 2.590 segundos)

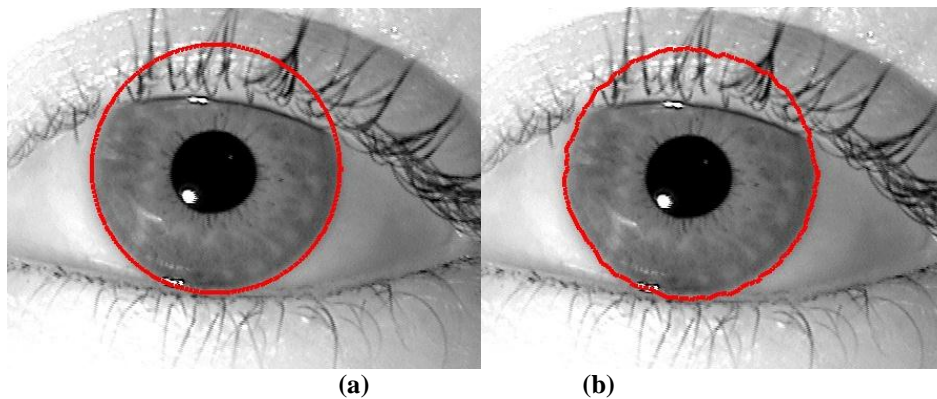


Figura 7.26: 288615_245178_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 15$, $\beta = 1$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecución = 2.691 segundos)

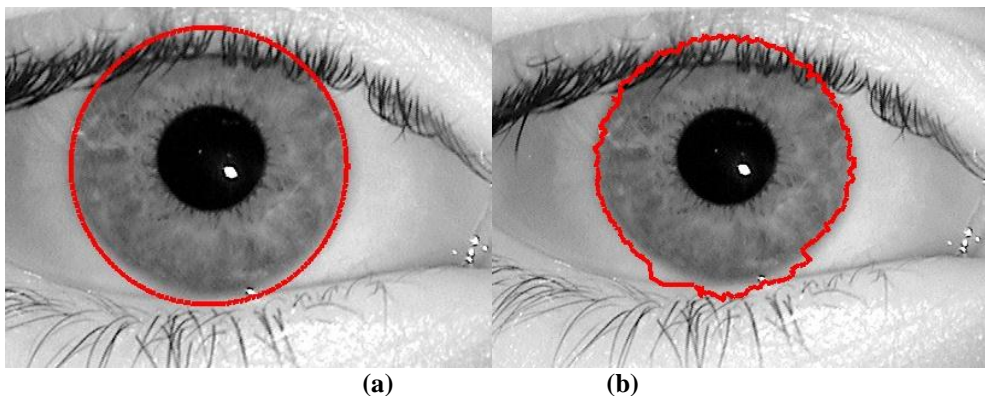


Figura 7.27: 290821_240036_D.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 0.55$, $\beta = 0$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecución = 2,711 segundos)

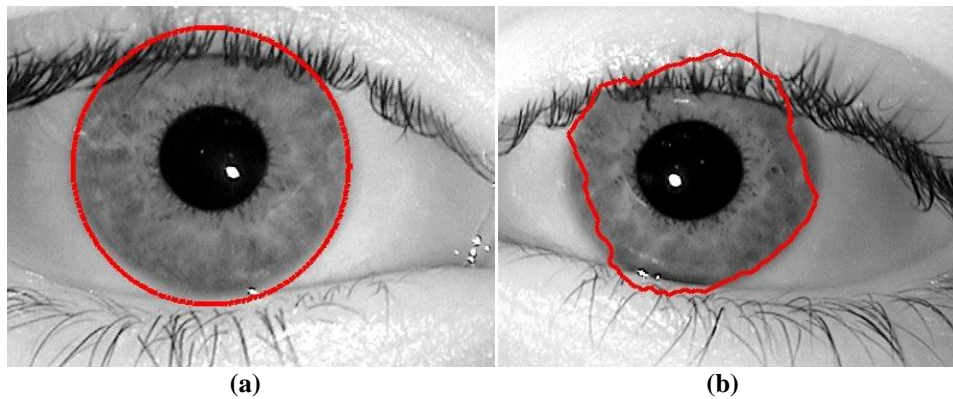


Figura 7.28: 290821_240036_D.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 15$, $\beta = 0$, $\gamma = 2$, iteraciones = 50. Tiempo de ejecución = 2,765 segundos)

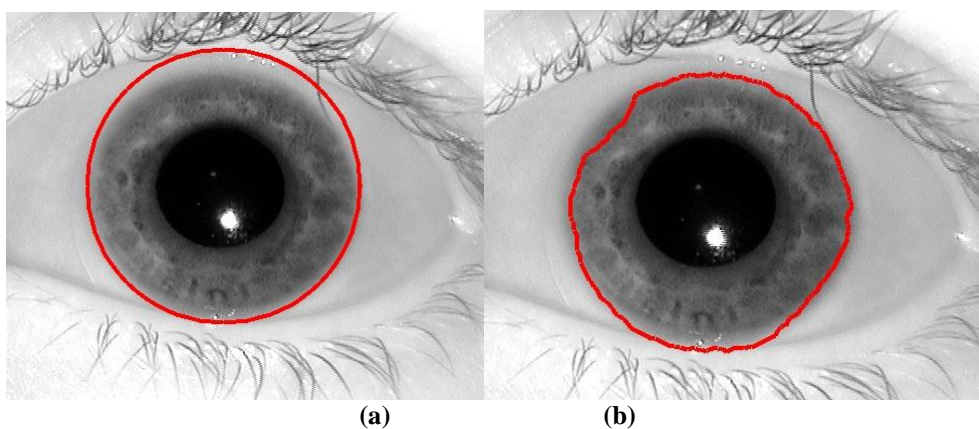


Figura 7.29: 290880_245443_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 5$, $\beta = 1$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecución = 2.849 segundos)

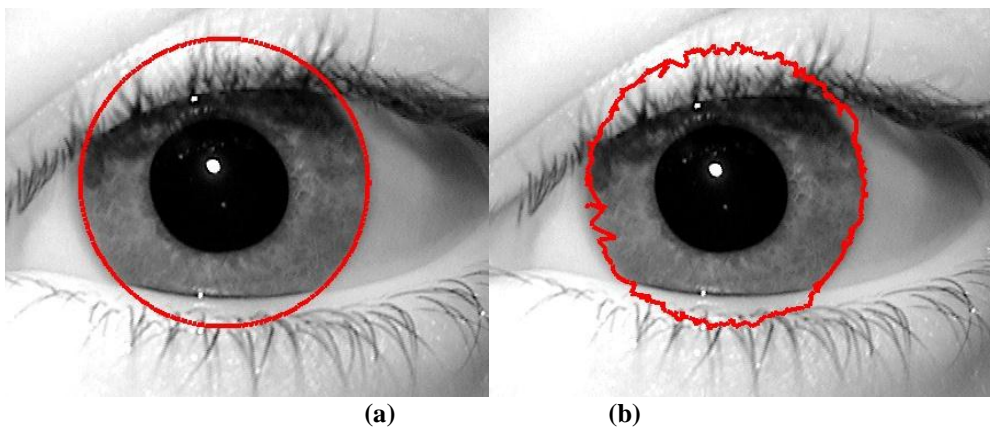


Figura 7.30: 290880_245443_I.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 0.55$, $\beta = 1$, $\gamma = 1$, iteraciones = 50. Tiempo de ejecución = 2.982 segundos)

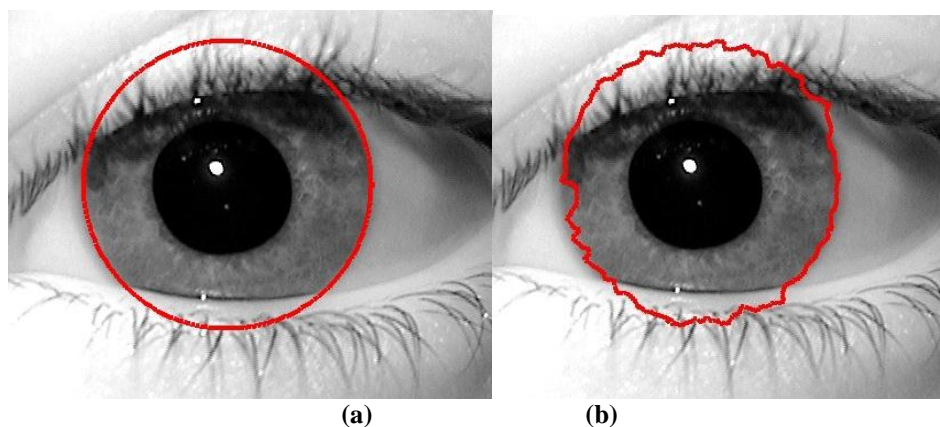


Figura 7.31: 290880_245443_1.bmp (a) imagen original con los contornos iniciales. (b) imagen tras aplicar el snake. ($\alpha = 5.55$, $\beta = 0$, $\gamma = 2$, iteraciones = 50. Tiempo de ejecuci3n = 2.599 segundos)

A continuaci3n podemos ver un breve an3lisis de los datos obtenidos:

Tiempo de ejecuci3n	
Media	2,801 seg.

Tabla 7.8: Tiempos de ejecuci3n

Debido a la gran variedad de las im3genes de entrada, se ha decidido realizar un estudio de diferentes umbrales. Para ello se han considerado un total de 500 im3genes elegidas aleatoriamente de la base de datos. A continuaci3n vemos una tabla del an3lisis realizado para cada una de las im3genes.

		<i>snake</i> - borde externo						
Alpha		5	9	15	5	9	12	5
Beta		0	0	0	1	2	2	1
Gamma		1	1	1	1	1	2	2
num. de iter		50	50	50	50	50	50	50
Borde	Adecuado	62%	67%	71%	63%	69%	58%	56%
	Posibles errores	36%	30%	27%	35%	30%	39%	41%
	Malo	2%	3%	2%	2%	1%	3%	3%

Tabla 7.9: An3lisis de los resultados despu3s ejecutar el *snake* para el borde externo.

7.2.4. Conclusi3n

Los *snakes* son adaptables a diferentes entornos. Sus par3metros permiten que sea adaptado al reconocimiento de bordes de diferentes caracter3sticas, dependiendo de las necesidades particulares de una aplicaci3n dada.

Para el borde interno, los resultados logrados son plenamente satisfactorios con los par3metros adecuados. Estos par3metros permiten ajustar el algoritmo a distintos tipos de im3genes, especialmente a im3genes con formas circulares. Para nuestro caso, tras probar con diferentes valores para estos par3metros, se ha realizado un estudio para conseguir los valores m3s adecuados. Estos valores han sido elegidos tras valorar sus posibilidades desde el punto de vista te3rico. Se ha intentado elegir un valor bajo de

beta para que el contorno no tenga muchos vértices o esquinas. Se ha probado con distintos valor de alpha, ya que si observamos las imágenes obtenidas tras Canny, los bordes interiores suelen estar muy bien definidos, por lo que será más complicado que encoja.

Aunque estos parámetros hacen que el *snake* se comporte de una forma u otra, para el caso del borde interior, vemos que tenemos una tasa de acierto muy elevada para diferentes valores de parámetros, siguiendo unos valores base. La tasa de acierto ha llegado hasta el 92%. Esto es debido, a que en la fase de pre-procesado, en la de Canny y GVF, conseguimos muy buenos resultados, lo que nos permite pintar un contorno inicial muy apropiado. Los resultados que nos dan son muy satisfactorios.

Para el borde externo, los resultados no son plenamente satisfactorios. Se consiguen muy buenos resultados, pero no se ha tenido en cuenta las pestañas a la hora de diseñar el algoritmo para esta parte. El no tener en cuenta las pestañas hace que en todas las imágenes en las que los ojos están medio cerrado, no conseguimos resultados plenamente satisfactorios. Son muy buenos, pero al no tener en cuenta estos aspectos, hace que se eleven las tasas de error. Se ha intentado incrementar el valor de alpha para que el *snake* tienda a encoger, pero no en todos los casos se consigue evitar las pestañas.

Para solucionar este inconveniente, también se ha intentado implementar otra forma de obtener los contornos iniciales. En lugar de obtenerlos a partir del pre-procesado, lo podemos pintar nosotros sobre la imagen (de forma que sea una elipse, o un recorrido cerrado de puntos). De esta forma cada imagen se convierte en un caso individual, ya que no hemos sido capaces de encontrar ningún patrón para hacerlo de forma continua, y automática sobre cada imagen.

Los tiempos de ejecución también se han analizado y se han considerado muy satisfactorios. En comparación con otros lenguajes u otro hardware, nuestros tiempos se pueden considerar ideales para poder implementar un sistema real.

Parte V

Conclusiones y líneas futuras

Capítulo 8

Conclusiones y líneas futuras

En el presente proyecto se detallan todas y cada una de las fases llevadas a cabo para la elaboración de un algoritmo de segmentación de iris ocular. Para ello ha sido necesario un estudio a fondo de diversas técnicas de procesado de imágenes. Ha sido pues un interesante y largo aprendizaje, además de suponer una gran oportunidad de aprender una materia completamente nueva para mí, la Biometría y sus diferentes usos.

Para el desarrollo del algoritmo ha sido necesario internarse en el interesante y complicado estudio de los contornos activos, de los diferentes modelos existentes para lograr implementar un algoritmo que proporcionase resultados aceptables en un tiempo asumible con los recursos que tenemos a nuestra disposición. El resultado ha sido plenamente satisfactorio. Aun así de este proyecto se pueden sacar muchas propuestas de trabajos futuros, que mejoren, amplíen o corrijan el sistema presentado. Algunas de estas propuestas se detallan en este capítulo.

8.1. Conclusiones

El objetivo principal de este proyecto ha consistido en implementar un sistema que sea capaz de detectar los bordes internos y externos del iris. Hasta ahora los sistemas actuales tenían algunas dificultades que no eran capaces de superar. En el diseño de este sistema se tienen en cuenta estas irregularidades que presenta el ojo humano como: surcos, reflejos por lo que no se esperaba encontrar problemas con estas dificultades. Tener en cuenta estas irregularidades permite realizar un sistema más robusto. El uso de contornos activos, frente a otras técnicas más robustas, pero incapaces de ofrecer soluciones que se adapten a todo los tipos de imágenes, como lentillas, reflejos, ojos asiáticos, manchas, etc. Los métodos usados la fecha, solían aproximar el iris humano por un círculo o elipse, mientras que con los contornos activos, podemos observar como en los ejemplos anteriores se deforman hasta adaptarse totalmente a los bordes internos o externos del iris. La tasa de acierto de los métodos usados hasta la fecha ronda el 85%, por lo que no descartemos considerar este trabajo con una opción de mejora y estudio.

Uno de los mayores problemas antes de desarrollar este proyecto era los tiempos de ejecución que tenían los desarrollos de contornos activos en lenguajes como Matlab. Se sabe que si se ejecuta parte de este código en C++ se mejoran muchos. Sin embargo, en este proyecto se ha demostrado que los tiempos de ejecución son muy aceptables. Con unos tiempos de cómputo para cada uno de los bordes, en la máquina descrita anteriormente, no suele superar los 3 segundos.

Finalmente, como conclusión de este proyecto, se han constatado las enormes dificultades que conlleva el procesado de imágenes, sabiendo la dificultad que lleva analizar la gran capacidad de procesado necesaria para encontrar un ojo en una imagen, en la cual, básicamente, no hay nada más que un ojo. Es un simple ejemplo de lo lejos que puede llegar esta materia, y que ya podemos encontrar aplicaciones asombrosas, que gracias a los sistemas actuales podemos conseguir muy buenos resultados

Los resultados logrados en el presente proyecto a partir de los medios disponibles son muy satisfactorios. Cabe destacar que al no tener en cuenta las pestañas desde el inicio del diseño, ha disminuido nuestra tasa de acierto para el borde anterior. A pesar de esto, para el borde interno se consiguen resultados muy buenos, y el tiempo de ejecución es muy bueno también.

8.2. Líneas futuras

Son numerosas las aplicaciones que pueden surgir de un sistemas biométrico como el aquí presentado. Es por ello, y porque aún queda mucho por desarrollar, que todo sistema se puede mejorar. En este pequeños apartado vamos a tratar de que sistemas o que mejoras podrían aumentar el rendimiento de nuestro sistema.

Un complemento que aumentaría el rendimiento del sistema sería aquel capaz de desechar fotografías que no ofrezcan una calidad suficiente para el pre-procesado, ya fuese por falta de nitidez, apertura del ojo, brillos, etc.

Otro complemento que aumentaría la eficacia del sistema, sería encontrar o conseguir ajustar correctamente el método de ajuste automático para imágenes binarias. Este método que ha sido descartado en nuestro sistema, puede aportar muchas facilidades a la hora de buscar los bordes que realmente nos interesa, y eliminar los bordes, que a la hora de detectar, nos molesten.

Otra línea futura, es la posibilidad de extrapolar el método de los contornos activos a cualquier caso que necesite detección de objetos. Este método se ajusta a cualquier imagen, y da muy buenos resultados en imágenes circulares.

Una línea futura muy interesante, sería la posibilidad de utilizar otro tipo de *snake*, como *snake ballons*, de los que no se ha hablado en este proyecto para detectar el borde exterior. Hemos visto que las pestañas u ojos medio cerrados pueden presentar problemas a nuestro sistema. La forma en la que actúan estos *snakes* podrían ser ideales para estos bordes.

Finalmente, la inclusión de este algoritmo en un sistema completo, y real de reconocimiento de iris.

Parte VI

Presupuesto

Capítulo 9

Presupuesto

9.1. Introducción

El proyecto con título: “Mejora de un algoritmo de segmentación de iris mediante contornos activos”, ha sido descrito a lo largo de los capítulos anteriores, junto con sus pruebas. A continuación podemos ver un desglose de costes de personal, material y costes totales.

UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor: David Alcón Corral

2.- Departamento: Tecnología Electrónica

3.- Descripción del Proyecto:

- Título: Mejora de un algoritmo de segmentación de iris mediante contornos activos.
- Duración (meses): 10 meses
- Tasa de costes Indirectos: **20%**

4.- Presupuesto total del Proyecto (valores en Euros):

29.442,00 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F.	Categoría	Dedicación (hombres mes)	Coste hombre mes	Coste (Euro)	Firma de conformidad
Alcón Corral, David	-	Ingeniero	10	2.400,00	24.000,00	
Total					24.000,00	

EQUIPOS					
Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
Acer Aspire 5943G, i7-720QM	930,00	100	10	60	155,00
M Visual Studio 2008 Pro	600,00	100	10	60	100,00
Emgu CV license single dev.	180,00	100	10	60	30,00
		100		60	0,00
Total					285,00

OTROS COSTES DIRECTOS DEL PROYECTO		
Descripción	Empresa	Costes imputable
Material oficina		20,00
Gastos impresión		30,00
Gastos encuadernación		200,00
Total		250,00

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	24.000
Amortización	285
Costes de funcionamiento	250
Costes Indirectos	4.907
Total	29.442

El presupuesto total de este proyecto asciende a la cantidad de veintitrés mil seiscientos con ochenta y dos euros.

Leganés a 10 de Julio de 2012
 El ingeniero proyectista
 Fdo: David Alcón Corral

Glosario

ANSI	American National Standards Institute SC17 Sub-Comité 17 (SC17)
CBEFF	CommonBiometric Exchange File Format
CEN	Comité Europeo de Normalización
JTC1	Joint Technical Committee on Information Technology
IEC	International Electrotechnical Commission
INCITS	Inter National Committee for Information Technology Standards
ISO	International Organization for Standardization
NIST	National Institute of Standards and Technology

Bibliografía

[i] Orwell, George (2009). *1984*. Colección: Áncora y Delfín. Barcelona: Ediciones Destino. [ISBN 978-84-233-4165-8](#).

- [1] Artículo Wikipedia: Biometría. Consultado el 1 de Junio de 2012
- [2] imagen http://www2.ing.puc.cl/~iing/ed429/sistemas_biometricos.htm
- [3] tabla “Biometrics: Personal Identification in Networked Society” de Anil K. Jain, Ruud Bolle, SharathPankanti, Ed. Springer, 1999
- [4] Imagen <http://commons.wikimedia.org/wiki/File:Eyesection-es.svg?uselang=es>
- [5] Daugman, John, “Probing the Uniqueness and randomness of iris codes: results from 200 billion iris pair comparisons”, IEEE Proceedings 2006
- [6] Artículos de wikipedia: segmentation (image processing) y procesamiento digital de imágenes. Consultado el 27 de Diciembre de 2010
- [7] Lindeberg, Tony (2001), "Edge detection", in Hazewinkel, Michiel, Encyclopedia of Mathematics, Springer, ISBN 978-1556080104
- [2.5.4] T. Lindeberg (1998) "Edge detection and ridge detection with automatic scale selection", International Journal of Computer Vision, 30, 2, pages 117--154.
- [8] Artículo Wikipedia: Binary Image, Otsu method, Canny, Sobel operator, Prewitt operator
- [9] 09gr820 "Canny Edge Detection", March 23, 2009
- [10] Aryn Poonawala “Active Contours Model”. EE 264. Instructor: Dr. Peyman Milanfar. . <http://www.soe.ucsc.edu/classes/ee264/Winter02/aryn.ppt>
- [11] Luis Criado “Identificación de la papila usando el algoritmo de contornos activos”. Departamento de Inteligencia Artificial (UNED). Año 2002. <http://www.criado.info/descarga/doctorado/papila%20criado.pdf>
- [12] “Snakes: Active Contour Models” Michael Kass, Andrew Witkin, Demetri Terzopoulos. International Journal of Computer Vision, 321-331 (1988).
- [13] Marcos Martín “Contornos Activos”. 15 de Mayo de 2002. <http://poseidon.tel.uva.es/~carlos/ltif10001/contactivos.pdf>.
- [14] Artículos de wikipedia: Active Contour Model. Consultado el 27 de Diciembre 2010.
- [15] Chenyang Xu and Jerry L. Prince “Gradient Vector Flow: A New External Force for Snakes”, IEEE, pp 66-71, Junio 1997.
- [16] Tesis Doctoral de Rafael Verdú Monedero “Formulación de los contornos activos en el dominio de la frecuencia y análisis de convergencia en segmentación de imagen”. Departamento de Tecnologías de la Información y las Comunicaciones. Directores: Dr. Juan Morales Sánchez y Dr. Luis Weruaga Prieto. Año 2005
- [17] Zoltan Kato “Snakes: Active Contours”. <http://www.inf.u-szeged.hu/~kato/teaching/emm/02-Snake.pdf>
- [18] Francisco López Valverde y Nicolás Guil Mata “Localización de vasos sanguíneos en mamografías mediante contornos activos”. Revista Salud.com, Vol 3, Número 9 (2007) – I Trimestre.
- [19] Xu, C. y J. L. Prince, “Snakes, shapes, and gradient vector flow”, IEEE Transactions on Image Processing, Vol 7, no. 3, Marzo 1998.

[20] Chenyang Xu and Jerry L. Prince “Gradient Vector Flow Deformable Models”. Handbook of Medical Imaging, Editor: Isaac Bankman, Academic Press, Septiembre 2000.

[21] C. Gout, C. Le Guyader “Geodesic active contour under geometrical conditions: Theory and 3D Applications”. Numerical Algorithms, to appear 2008.